

## Assignment – 2

Name – Sayak Sen

Enrollment No – 2023CSB047

Subject – Computer Networks Lab

1. Analyse the packets (across all layers) exchanged with your computer while executing the following commands: (i) ping (ii) traceroute (iii) dig (iv) arp (v) wget.

(i) ping

Wireshark packet capture showing ICMP Echo (ping) requests. The capture is from interface wlp2s0. The packet list shows 14 packets, all ICMP Echo (ping) requests from 10.74.218.242 to 1.1.1.1. The packet details pane shows the ICMP header and the IP header. The packet bytes pane shows the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
5	4.825806324	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=1/256, tt
6	4.878233944	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=1/256, tt
7	5.826354637	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=2/512, tt
8	5.870431250	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=2/512, tt
9	6.826491184	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=3/768, tt
10	6.869670539	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=3/768, tt
54	7.827061346	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=4/1024, t
55	7.975876998	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=4/1024, t
56	8.828093992	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=5/1280, t
57	8.876550409	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=5/1280, t
58	9.828251999	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=6/1536, t

Total Length: 84  
Identification: 0xb81c (47132)  
010. .... = Flags: 0x2, Don't fragment  
...0 0000 0000 0000 = Fragment Offset: 0  
Time to Live: 64  
Protocol: ICMP (1)  
Header Checksum: 0x9b4e [validation disabled]  
[Header checksum status: Unverified]  
Source Address: 10.74.218.242  
Destination Address: 1.1.1.1  
[Stream index: 1]  
Internet Control Message Protocol

Wireshark packet capture showing ICMP Echo (ping) requests. The capture is from interface wlp2s0. The packet list shows 14 packets, all ICMP Echo (ping) requests from 10.74.218.242 to 1.1.1.1. The packet details pane shows the ICMP header and the IP header. The packet bytes pane shows the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
5	4.825806324	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=1/256, tt
6	4.878233944	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=1/256, tt
7	5.826354637	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=2/512, tt
8	5.870431250	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=2/512, tt
9	6.826491184	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=3/768, tt
10	6.869670539	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=3/768, tt
54	7.827061346	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=4/1024, t
55	7.975876998	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=4/1024, t
56	8.828093992	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=5/1280, t
57	8.876550409	1.1.1.1	10.74.218.242	ICMP	98	Echo (ping) reply id=0x1c68, seq=5/1280, t
58	9.828251999	10.74.218.242	1.1.1.1	ICMP	98	Echo (ping) request id=0x1c68, seq=6/1536, t

Total Length: 84  
Identification: 0x0000 (0)  
010. .... = Flags: 0x2, Don't fragment  
...0 0000 0000 0000 = Fragment Offset: 0  
Time to Live: 54  
Protocol: ICMP (1)  
Header Checksum: 0x5d6b [validation disabled]  
[Header checksum status: Unverified]  
Source Address: 1.1.1.1  
Destination Address: 10.74.218.242  
[Stream index: 1]  
Internet Control Message Protocol

The ping command uses the **ICMP (Internet Control Message Protocol)** at the network layer.

In Wireshark, filtering with icmp shows two packets for each ping:

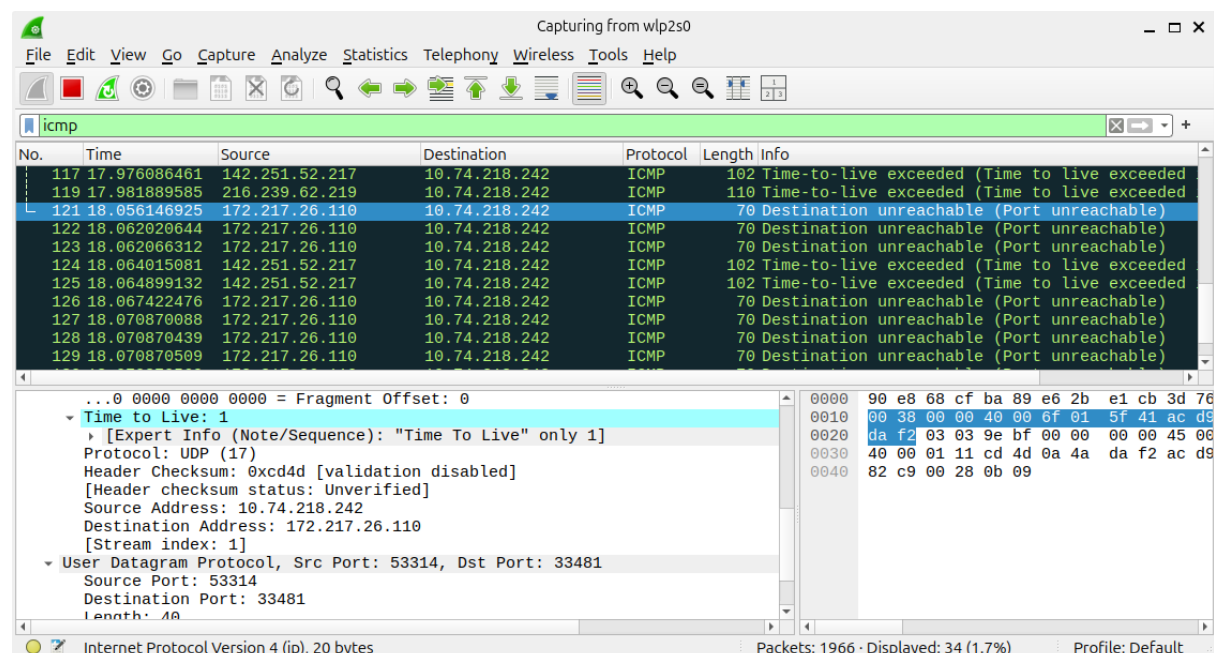
- **ICMP Echo Request (Type 8)** from the source to destination
- **ICMP Echo Reply (Type 0)** from destination back to source

Encapsulation observed:

- **Ethernet Layer:** Source and Destination MAC addresses
- **IP Layer:** Source IP (10.74.218.242) and Destination IP (1.1.1.1)
- **ICMP Layer:** Type, Code, Checksum, Sequence Number

This confirms ping checks connectivity and round-trip time without using TCP/UDP.

## (ii) traceroute



Behavior seen:

- Packets are sent with TTL = 1, 2, 3...
- Each router where TTL becomes 0 sends back ICMP Time Exceeded
- When the destination is reached, it sends ICMP Port Unreachable. Thus, traceroute reveals each router (hop) along the path to the destination.

### (iii) dig

The screenshot shows a Wireshark capture of DNS traffic. The packet list pane displays several DNS packets. The selected packet (No. 391) is a DNS Standard query response from 10.74.218.188 to 10.74.218.242. The packet details pane shows the transaction ID 0xe801, flags 0x8180, and a single query response for drivefrontend-pa.clients6.google.com. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
351	10.74.218.242	10.74.218.188	DNS	96	Standard query 0xe801 HTTPS drivefrontend-pa.clients6.google.com	
399	10.74.218.242	10.74.218.188	DNS	96	Standard query 0x5217 AAAA drivefrontend-pa.clients6.google.com	
232	10.74.218.242	10.74.218.188	DNS	96	Standard query 0x6897 A drivefrontend-pa.clients6.google.com	
391	10.74.218.188	10.74.218.242	DNS	146	Standard query response 0xe801 HTTPS drivefrontend-pa.clients6.google.com	
142	10.74.218.188	10.74.218.242	DNS	112	Standard query response 0x6897 A drivefrontend-pa.clients6.google.com	
187	10.74.218.188	10.74.218.242	DNS	124	Standard query response 0x5217 AAAA drivefrontend-pa.clients6.google.com	
307	10.74.218.242	10.74.218.188	DNS	74	Standard query 0x9664 A www.google.com	
148	10.74.218.188	10.74.218.242	DNS	90	Standard query response 0x9664 A www.google.com A 142.250.193	
3110	10.74.218.242	10.74.218.188	DNS	86	Standard query 0xeb4f HTTPS waa-pa.clients6.google.com	
3797	10.74.218.242	10.74.218.188	DNS	86	Standard query 0x78c5 AAAA waa-pa.clients6.google.com	
2864	10.74.218.242	10.74.218.188	DNS	86	Standard query 0x721f A waa-pa.clients6.google.com	

Transaction ID: 0xe801  
Flags: 0x8180 Standard query response, No error  
Questions: 1  
Answer RRs: 0  
Authority RRs: 1  
Additional RRs: 0  
Queries  
drivefrontend-pa.clients6.google.com: type HTTPS, class IN  
Authoritative nameservers  
google.com: type SOA, class IN, mname ns1.google.com  
[Request In: 3]  
[Time: 40.765840 milliseconds]

### Observed:

- DNS Query packet from my system to DNS server (UDP port 53)
- DNS Response packet containing the A record (IPv4 address)

This shows how domain names are translated into IP addresses.

### (iv) arp

The screenshot shows a Wireshark capture of ARP traffic. The packet list pane displays several ARP packets. The selected packet (No. 71) is an ARP request from e6:2b:e1:cb:3d:76 to e6:2b:e1:cb:3d:76. The packet details pane shows the hardware type Ethernet (1), protocol type IPv4 (0x0800), and the request details. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
70	9.054147320	AzureWaveTec_cf:ba:...	e6:2b:e1:cb:3d:76	ARP	42	who has 10.74.218.188? Tell 10.74.218.242
71	9.060075068	e6:2b:e1:cb:3d:76	AzureWaveTec_cf:ba:...	ARP	42	who has 10.74.218.242? Tell 10.74.218.188
72	9.060093214	AzureWaveTec_cf:ba:...	e6:2b:e1:cb:3d:76	ARP	42	10.74.218.242 is at e6:2b:e1:cb:3d:76
73	9.068556100	e6:2b:e1:cb:3d:76	AzureWaveTec_cf:ba:...	ARP	42	10.74.218.188 is at e6:2b:e1:cb:3d:76
141	42.555137489	e6:2b:e1:cb:3d:76	AzureWaveTec_cf:ba:...	ARP	42	who has 10.74.218.242? Tell 10.74.218.188
142	42.555152068	AzureWaveTec_cf:ba:...	e6:2b:e1:cb:3d:76	ARP	42	10.74.218.242 is at e6:2b:e1:cb:3d:76
186	56.679174406	AzureWaveTec_cf:ba:...	e6:2b:e1:cb:3d:76	ARP	42	who has 10.74.218.188? Tell 10.74.218.242
187	56.686609710	e6:2b:e1:cb:3d:76	AzureWaveTec_cf:ba:...	ARP	42	10.74.218.188 is at e6:2b:e1:cb:3d:76
331	101.748422261	e6:2b:e1:cb:3d:76	AzureWaveTec_cf:ba:...	ARP	42	who has 10.74.218.242? Tell 10.74.218.188
332	101.748440347	AzureWaveTec_cf:ba:...	e6:2b:e1:cb:3d:76	ARP	42	10.74.218.242 is at e6:2b:e1:cb:3d:76
425	162.653892063	AzureWaveTec_cf:ba:...	e6:2b:e1:cb:3d:76	ARP	42	who has 10.74.218.188? Tell 10.74.218.242

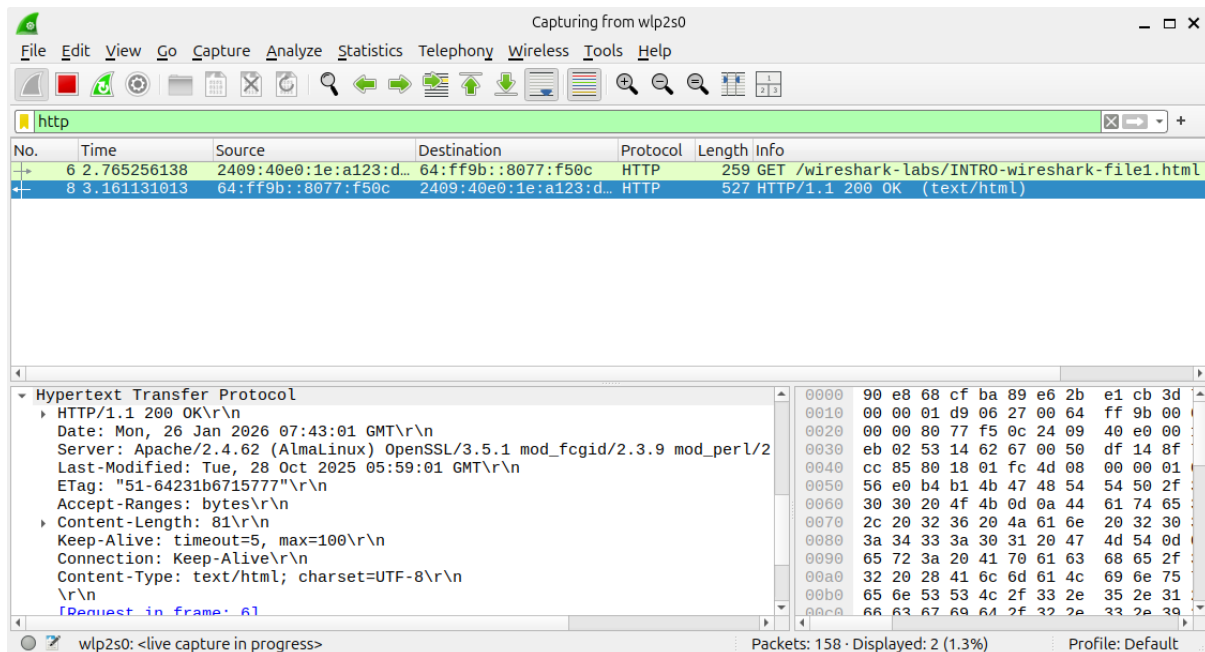
Frame 71: Packet, 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on Ethernet II, Src: e6:2b:e1:cb:3d:76 (e6:2b:e1:cb:3d:76), Dst: AzureWaveTec\_cf: Address Resolution Protocol (request)  
Hardware type: Ethernet (1)  
Protocol type: IPv4 (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: request (1)  
Sender MAC address: e6:2b:e1:cb:3d:76 (e6:2b:e1:cb:3d:76)  
Sender IP address: 10.74.218.188  
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)  
Target IP address: 10.74.218.242

## Observed:

- ARP Request (Broadcast): “Who has <gateway IP>?”
- ARP Reply (Unicast): “<gateway IP> is at <MAC address>”

This demonstrates that before sending packets outside the network the system must know the gateway’s MAC address.

## (v) wget

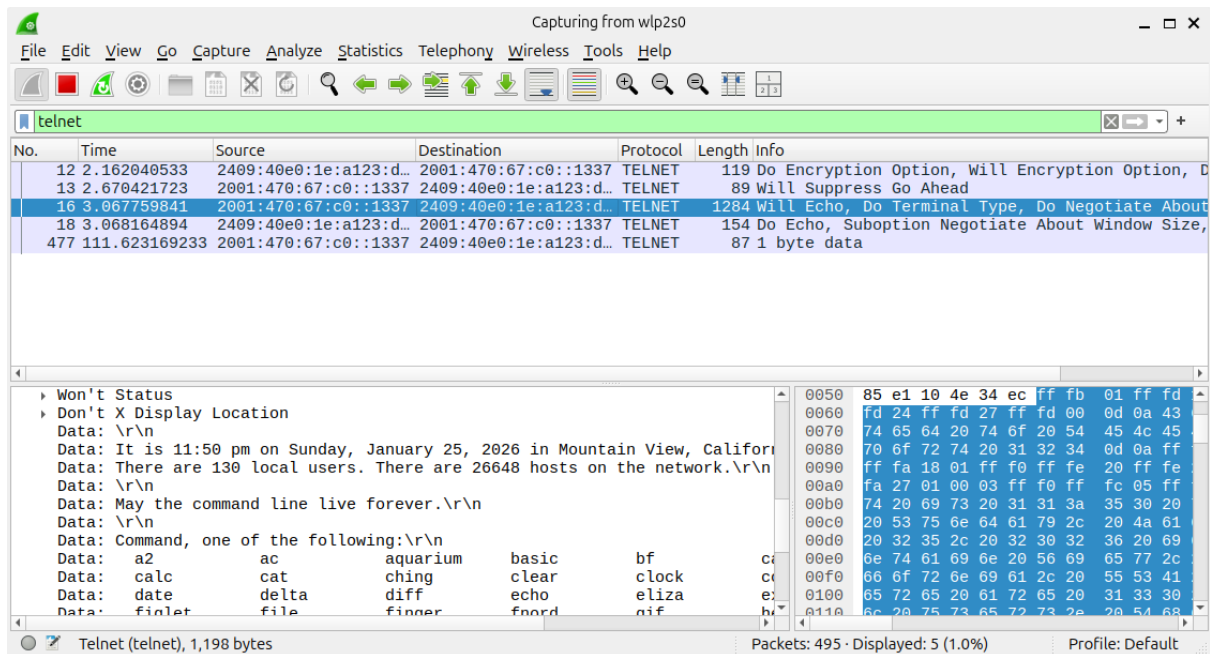


## Process observed:

1. TCP 3-way handshake
2. HTTP GET request sent to server
3. HTTP 200 OK response received
4. File data transferred
5. TCP connection closed

The HTTP data is visible in plain text, showing headers and content.

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?



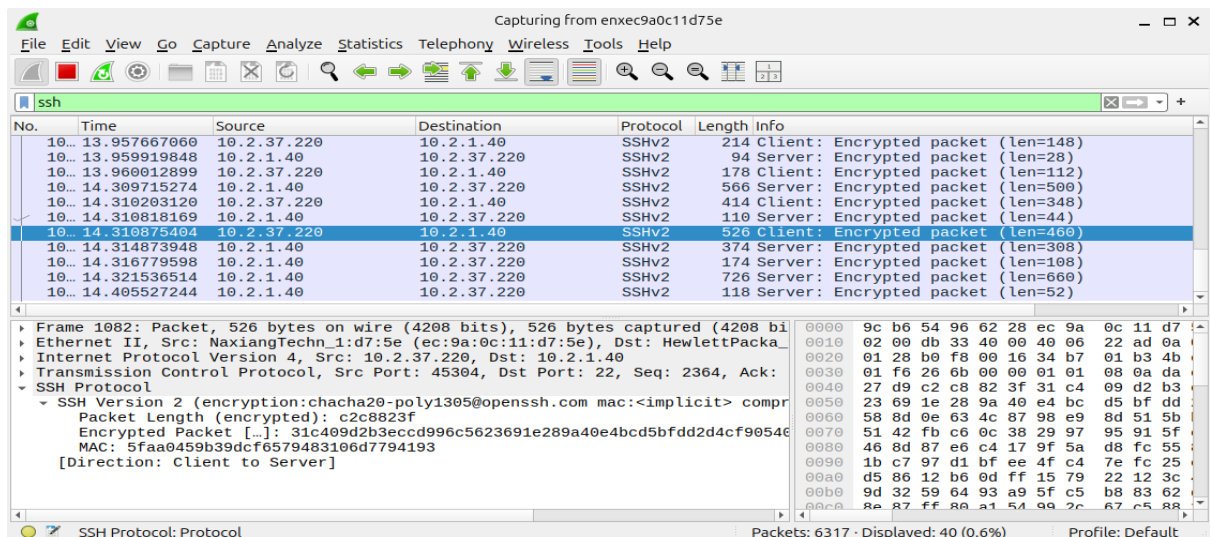
**Telnet uses TCP port 23 and does not encrypt data.**

**In Wireshark:**

- Filter: telnet or tcp.port == 23
- Application layer data is fully visible
- Username, password, and typed commands can be read in plain text

**Observation: Telnet is insecure because data is transmitted without encryption.**

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





SSH uses TCP port 22 and provides encrypted communication.

Observed:

- SSH handshake packets
- All application data appears as encrypted payload
- No readable usernames or commands

Observation: Unlike Telnet, SSH secures data using encryption.

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.

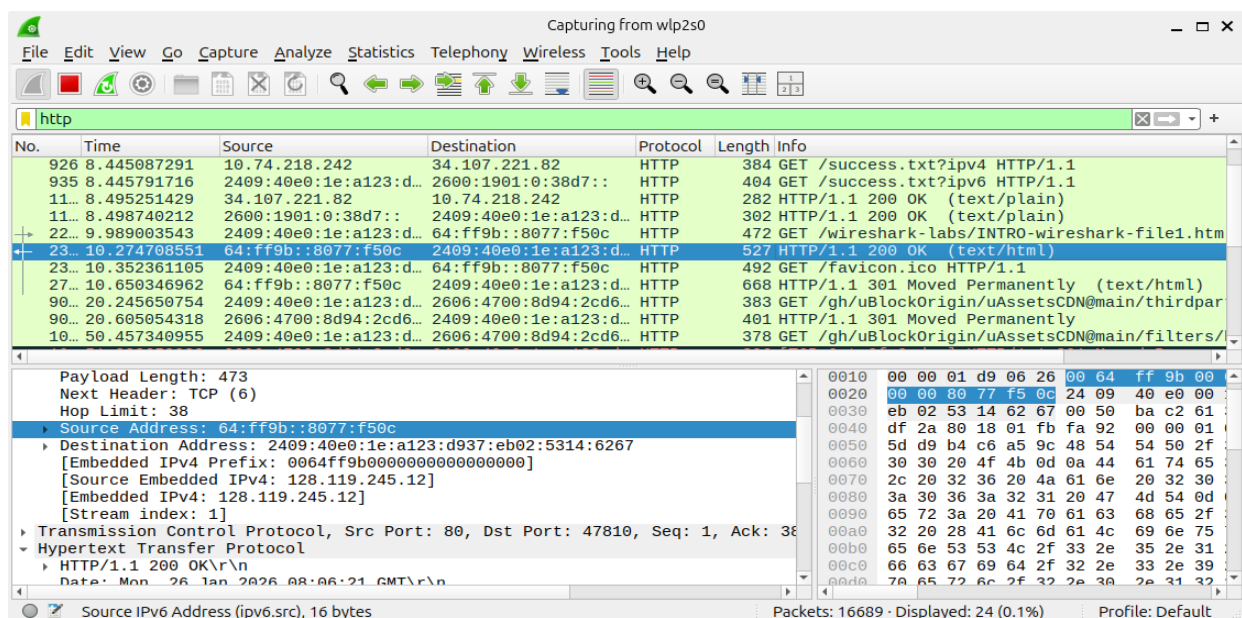
Answer the following from the captured packets:

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.



### (a) Time between HTTP GET and HTTP OK

By comparing timestamps:

- Time when HTTP GET was sent
- Time when HTTP/1.1 200 OK was received

The difference gives the server response time which is 0.05 seconds.

### (b) IP Addresses

From the IP header:

- Source IP = 34.107.221.82
- Destination IP = 10.74.218.242

These can be verified from packet details in Wireshark.

## 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?

The screenshot shows the Wireshark interface with a packet capture from 'wlp2s0'. The packet list on the left shows several packets, with packet 57 selected. The packet details pane on the right shows the structure of the selected packet:

- Frame 1: Packet, 175 bytes on wire (1400 bits), 175 bytes captured (1400 bits)
- Ethernet II, Src: AzureWaveTec\_cf:ba:89 (90:e8:68:cf:ba:89), Dst: ServercomPri
- Internet Protocol Version 6, Src: 2405:201:800f:614d:a004:88e:f807:6411, Dst:
- Transmission Control Protocol, Src Port: 35906, Dst Port: 443, Seq: 1, Ack: 1,
- Transport Layer Security
  - [Stream index: 0]
  - TLSv1.2 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
    - Content Type: Application Data (23)
    - Version: TLS 1.2 (0x0303)
    - Length: 84
    - Encrypted Application Data: e1b04b20b46db522ec0625511bfd4810a5c64d7aee634f
    - [Application Data Protocol: Hypertext Transfer Protocol]

The packet bytes pane on the right shows the raw data in hexadecimal and ASCII.

**(a) Difference in Application Layer Protocol**

**Yes. The protocol is HTTPS, which uses TLS encryption, instead of plain HTTP.**

**(b) Difference from Previous HTTP Capture**

<b>Feature</b>	<b>HTTP (Task 4)</b>	<b>HTTPS (Task 5)</b>
<b>Encryption</b>	<b>No</b>	<b>Yes (TLS)</b>
<b>Data visibility</b>	<b>Visible</b>	<b>Encrypted</b>
<b>Credentials visible</b>	<b>Yes</b>	<b>No</b>
<b>Port</b>	<b>80</b>	<b>443</b>

**Only TLS handshake and encrypted application data packets are visible in Wireshark.**