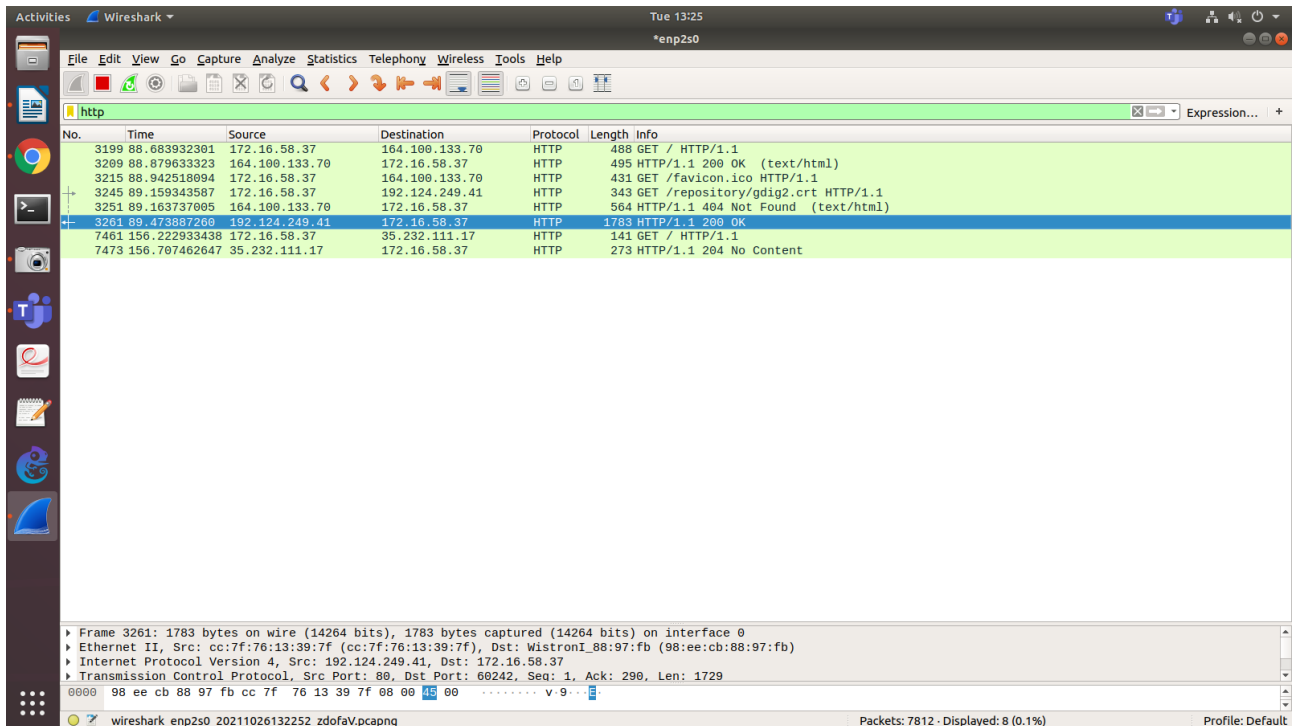# CN Lab 3

Name: Rhea Adhikari
Reg No: 190905156
Roll No: 23
Section: CSE-D

Q 3.1.   Retrieve web pages using HTTP. Use Wireshark to capture packets for analysis. Learn about most common HTTP messages. Also capture response messages and analyze them. During the lab session, also examine and analyze some HTTP headers.

I opened several websites on chrome and captured the packets displayed in wireshark.

I used the filter above just to find http request connections.

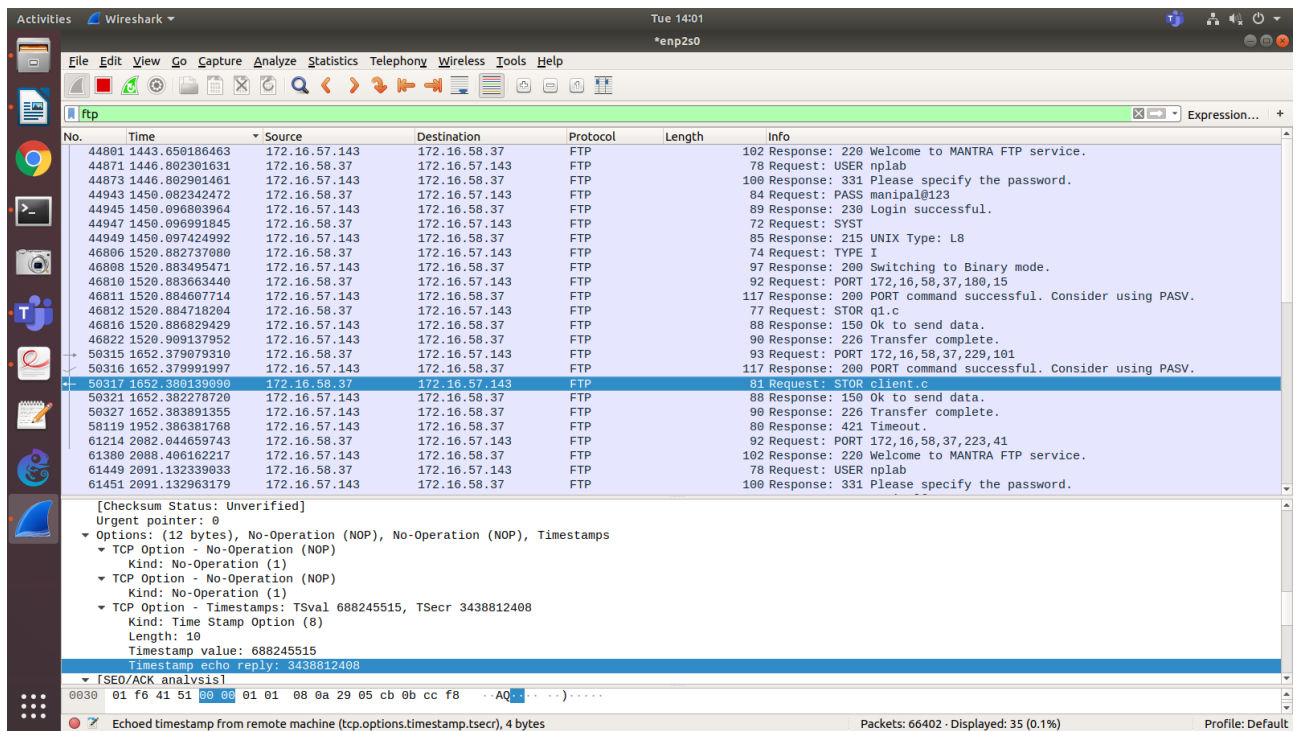**I observed two types of status codes which appeared**
   200 : OK
   204 : No Content
   404 : Not Found

```
▼ Internet Protocol Version 4, Src: 164.100.133.70, Dst: 172.16.58.37
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
   ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 550
      Identification: 0x6d6b (28011)
   ▶ Flags: 0x4000, Don't fragment
      Time to live: 63
      Protocol: TCP (6)
      Header checksum: 0xbc86 [validation disabled]
      [Header checksum status: Unverified]
      Source: 164.100.133.70
      Destination: 172.16.58.37
      [Time since reference of first frame: 89.100737005 seconds]
      Frame Number: 3251
      Frame Length: 564 bytes (4512 bits)
      Capture Length: 564 bytes (4512 bits)
      [Frame is marked: False]
      [Frame is ignored: False]
      [Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]
      [Coloring Rule Name: HTTP]
      [Coloring Rule String: http || tcp.port == 80 || http2]
         Via: HTTP/1.1 forward.http.proxy:3128\r\n
         Connection: close\r\n
         \r\n
         [HTTP response 1/1]
         [Time since request: 0.221218911 seconds]
         [Request in frame: 3215]
         [Request URI: http://kea.kar.nic.in/favicon.ico]
         File Data: 290 bytes
    ▼ Line-based text data: text/html (9 lines)
         <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
    0030  00 ed 75 8c 00 00 48 54  54 50 2f 31 2e 31 20 34   ··u···HT TP/1.1 4
    0040  30 34 20 4e 6f 74 20 46  6f 75 6e 64 0d 0a 44 61   04 Not F ound··Da
    0050  74 65 3a 20 54 75 65 2c  20 32 36 20 4f 63 74 20   te: Tue,  26 Oct
    0060  32 30 32 31 20 30 39 3a  32 35 3a 31 34 20 47 4d   2021 09: 25:14 GM
    0070  54 0d 0a 53 65 72 76 65  72 3a 20 41 70 61 63 68   T··Serve r: Apach
    0080  65 2f 32 2e 32 2e 31 35  20 28 52 65 64 20 48 61   e/2.2.15  (Red Ha
    0090  74 29 0d 0a 43 6f 6e 74  65 6e 74 2d 4c 65 6e 67   t)··Cont ent-Leng
    00a0  74 68 3a 20 32 39 30 0d  0a 43 6f 6e 74 65 6e 74   th: 290· ·Content
    00b0  2d 54 79 70 65 3a 20 74  65 78 74 2f 68 74 6d 6c   -Type: t ext/html
    00c0  3b 20 63 68 61 72 73 65  74 3d 69 73 6f 2d 38 38   ; charse t=iso-88
    00d0  35 39 2d 31 0d 0a 56 69  61 3a 20 48 54 54 50 2f   59-1··Vi a: HTTP/
    00e0  31 2e 31 20 66 6f 72 77  61 72 64 2e 68 74 74 70   1.1 forw ard.http
    00f0  2e 70 72 6f 78 79 3a 33  31 32 38 0d 0a 43 6f 6e   .proxy:3 128··Con
    ⬡  Hypertext Transfer Protocol (http), 220 bytes              Packets: 28741 · Displayed: 13 (0.0%)          Profile: Default
```

```
▼ Line-based text data: text/html (9 lines)
      <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">\n
      <html><head>\n
      <title>404 Not Found</title>\n
      </head><body>\n
      <h1>Not Found</h1>\n
      <p>The requested URL /favicon.ico was not found on this server.</p>\n
      <hr>\n
      <address>Apache/2.2.15 (Red Hat) Server at kea.kar.nic.in Port 80</address>\n
      </body></html>\n
```

Q 3.2   Use FTP to transfer some files, Use Wireshark to capture some packets. Show that FTP uses two separate connections: a control connection and a data-transfer connection. The data connection is opened and closed for each file transfer activity. Also show that FTP is an insecure file transfer protocol because the transaction is done in plaintext.

```
Student@project-lab:~$ ftp 172.16.57.143
Connected to 172.16.57.143.
220 Welcome to MANTRA FTP service.
Name (172.16.57.143:Student): nplab
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put xyz.txt
local: xyz.txt remote: xyz.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
6 bytes sent in 0.00 secs (76.0958 kB/s)
ftp>
```
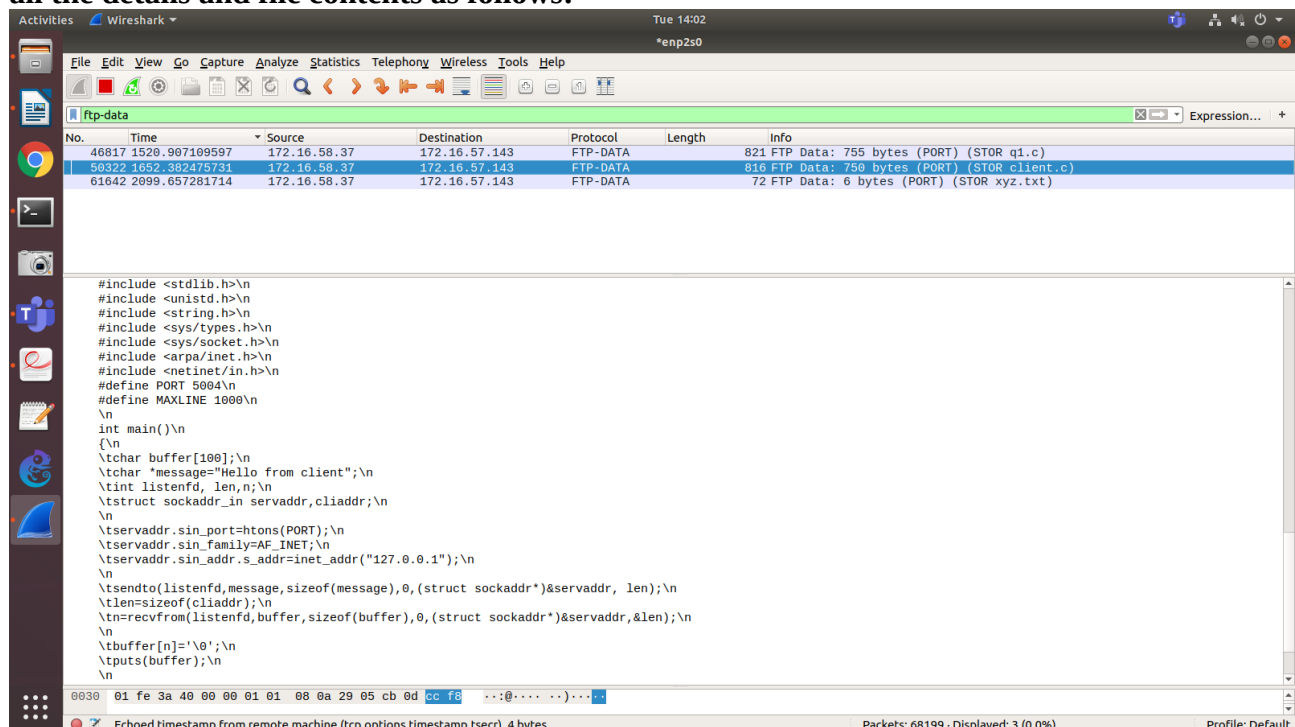
Data connection gets opened and closed.... control connection is consistent.

It is obvious that the communication is done in plain text and is insecure because we can see all the details and file contents as follows:

PORT 20-Data connection uses port 20 (TCP).

PORT 21- Control connection uses port number 21 (FTP)

Q 3.3   Analyze the behavior of the DNS protocol. In addition to Wireshark [Several network utilities are available for finding some information stored in the DNS servers. Eg.dig utilities (which has replaced nslookup). Set Wireshark to capture the packets sent by this utility.]

I used nslookup.io to search for swiggy.com and analysed the following windows below:s

4)
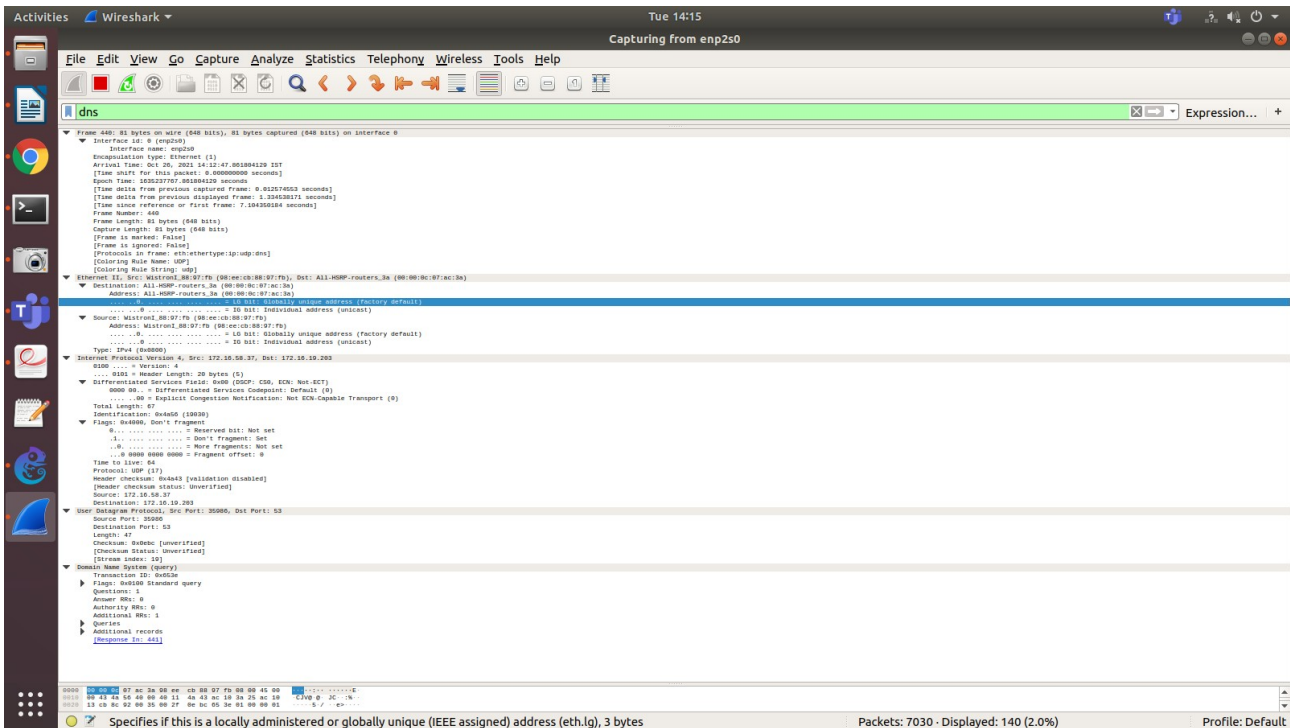
Firstly i did the following setup on GNS3
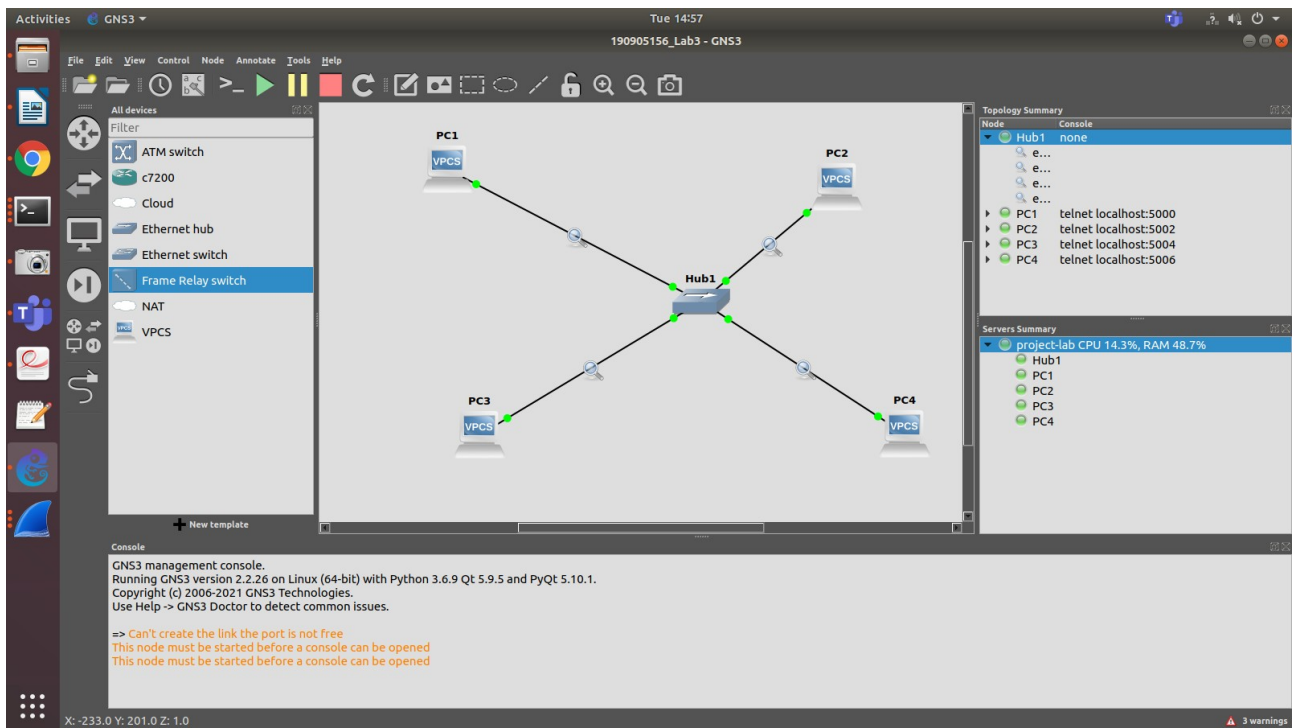
I used the following syntax to assign ip address to pc's

PCx> ip <ipaddress>

I connected 4 virtual Pcs to a Hub and assigned each of them different IP addresses as follows :
PC1- 10.0.0.1/24
PC2-10.0.0.2/24
PC3-10.0.0.3/24
PC4-10.0.0.4/24

pinged the following from pc1 to the rest of the pc's
pc1->pc2
pc1->pc3
pc1->pc4

MAC Address can be seen of each pc attached to the hub

```
PC3> show ip

NAME       : PC3[1]
IP/MASK    : 10.0.0.3/24
GATEWAY    : 10.0.0.254
DNS        :
MAC        : 00:50:79:66:68:02
LPORT      : 10010
RHOST:PORT : 127.0.0.1:10011
MTU        : 1500
```

```
PC4> show ip

NAME       : PC4[1]
IP/MASK    : 10.0.0.4/24
GATEWAY    : 10.0.0.254
DNS        :
MAC        : 00:50:79:66:68:03
LPORT      : 10014
RHOST:PORT : 127.0.0.1:10015
MTU        : 1500
```

Ethernet1

Ethernet2 (Next Page)

Activities Wireshark ▼   Tue 14:44

Capturing from Standard input [PC4 Ethernet0 to Hub1 Ethernet2]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter … <Ctrl-/>   Expression… +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.4? Tell 10.0.0.1 [ETHERNET FRAME CHECK SEQUENCE I…
2	0.000266	Private_66:68:03	Private_66:68:00	ARP	64	10.0.0.4 is at 00:50:79:66:68:03 [ETHERNET FRAME CHECK SEQUENCE I…
3	0.000969	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x76c5, seq=1/256, ttl=64 (reply in 4)
4	0.001269	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) request id=0x76c5, seq=1/256, ttl=64 (request in 3)
5	1.002359	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x77c5, seq=2/512, ttl=64 (reply in 6)
6	1.002700	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x77c5, seq=2/512, ttl=64 (request in 5)
7	2.003647	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x78c5, seq=3/768, ttl=64 (reply in 8)
8	2.003962	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x78c5, seq=3/768, ttl=64 (request in 7)
9	3.004721	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x79c5, seq=4/1024, ttl=64 (reply in 10)
10	3.004972	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x79c5, seq=4/1024, ttl=64 (request in 9)
11	4.006087	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x7ac5, seq=5/1280, ttl=64 (reply in 12)
12	4.006462	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x7ac5, seq=5/1280, ttl=64 (request in 11)
13	21.315992	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.3? Tell 10.0.0.1 [ETHERNET FRAME CHECK SEQUENCE IN…
14	21.316412	Private_66:68:02	Private_66:68:00	ARP	64	10.0.0.3 is at 00:50:79:66:68:02 [ETHERNET FRAME CHECK SEQUENCE I…
15	21.317008	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8bc5, seq=1/256, ttl=64 (reply in 16)
16	21.317331	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8bc5, seq=1/256, ttl=64 (request in 15)
17	22.318409	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8cc5, seq=2/512, ttl=64 (reply in 18)
18	22.318810	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8cc5, seq=2/512, ttl=64 (request in 17)
19	23.319665	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8dc5, seq=3/768, ttl=64 (reply in 20)
20	23.320054	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8dc5, seq=3/768, ttl=64 (request in 19)
21	24.320851	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8ec5, seq=4/1024, ttl=64 (reply in 22)
22	24.321207	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8ec5, seq=4/1024, ttl=64 (request in 21)
23	25.322093	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8fc5, seq=5/1280, ttl=64 (reply in 24)
24	25.322441	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8fc5, seq=5/1280, ttl=64 (request in 23)
25	57.112064	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.2? Tell 10.0.0.1 [ETHERNET FRAME CHECK SEQUENCE IN…
26	57.112648	Private_66:68:01	Private_66:68:00	ARP	64	10.0.0.2 is at 00:50:79:66:68:01 [ETHERNET FRAME CHECK SEQUENCE I…
27	57.114186	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xafc5, seq=1/256, ttl=64 (reply in 28)
28	57.114517	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xafc5, seq=1/256, ttl=64 (request in 27)
29	58.115471	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xb0c5, seq=2/512, ttl=64 (reply in 30)
30	58.116060	10.0.0.2	10.0.0.1	ICMP	98	Echo (ping) reply id=0xb0c5, seq=2/512, ttl=64 (request in 29)
31	59.116708	10.0.0.1	10.0.0.2	ICMP	98	Echo (ping) request id=0xb1c5, seq=3/768, ttl=64 (reply in 32)

▶ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface 0
▶ Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Address Resolution Protocol (request)

0000 ff ff ff ff ff ff 00 50 79 66 68 00 08 06 00 01   ·······P yfh····

Ready to load or capture   Packets: 36 · Displayed: 36 (100.0%)   Profile: Default

Ethernet3

Activities Wireshark ▼   Tue 14:47

Capturing from Standard input [PC2 Ethernet0 to Hub1 Ethernet3]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter … <Ctrl-/>   Expression… +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.0.0.3 (Request) [ETHERNET FRAME CHECK SEQU…
2	1.000841	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.0.0.3 (Request) [ETHERNET FRAME CHECK SEQUE…
3	2.001537	Private_66:68:02	Broadcast	ARP	64	Gratuitous ARP for 10.0.0.3 (Request) [ETHERNET FRAME CHECK SEQUE…
4	21.143869	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 10.0.0.4 (Request) [ETHERNET FRAME CHECK SEQUE…
5	22.143971	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 10.0.0.4 (Request) [ETHERNET FRAME CHECK SEQUE…
6	23.144451	Private_66:68:03	Broadcast	ARP	64	Gratuitous ARP for 10.0.0.4 (Request) [ETHERNET FRAME CHECK SEQUE…
7	152.283953	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.4? Tell 10.0.0.1 [ETHERNET FRAME CHECK SEQUENCE I…
8	152.284256	Private_66:68:03	Private_66:68:00	ARP	64	10.0.0.4 is at 00:50:79:66:68:03 [ETHERNET FRAME CHECK SEQUENCE I…
9	152.284912	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x76c5, seq=1/256, ttl=64 (reply in 9)
10	152.285270	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x76c5, seq=1/256, ttl=64 (request in 9)
11	153.286308	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x77c5, seq=2/512, ttl=64 (reply in 12)
12	153.286723	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x77c5, seq=2/512, ttl=64 (request in 11)
13	154.287597	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x78c5, seq=3/768, ttl=64 (reply in 14)
14	154.287964	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x78c5, seq=3/768, ttl=64 (request in 13)
15	155.288676	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x79c5, seq=4/1024, ttl=64 (reply in 16)
16	155.288920	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x79c5, seq=4/1024, ttl=64 (request in 15)
17	156.290036	10.0.0.1	10.0.0.4	ICMP	98	Echo (ping) request id=0x7ac5, seq=5/1280, ttl=64 (reply in 18)
18	156.290482	10.0.0.4	10.0.0.1	ICMP	98	Echo (ping) reply id=0x7ac5, seq=5/1280, ttl=64 (request in 17)
19	173.599939	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.3? Tell 10.0.0.1 [ETHERNET FRAME CHECK SEQUENCE IN…
20	173.600356	Private_66:68:02	Private_66:68:00	ARP	64	10.0.0.3 is at 00:50:79:66:68:02 [ETHERNET FRAME CHECK SEQUENCE I…
21	173.600950	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8bc5, seq=1/256, ttl=64 (reply in 22)
22	173.601277	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8bc5, seq=1/256, ttl=64 (request in 21)
23	174.602428	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8cc5, seq=2/512, ttl=64 (reply in 24)
24	174.602755	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8cc5, seq=2/512, ttl=64 (request in 23)
25	175.603616	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8dc5, seq=3/768, ttl=64 (reply in 26)
26	175.603999	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8dc5, seq=3/768, ttl=64 (request in 25)
27	176.604939	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8ec5, seq=4/1024, ttl=64 (reply in 28)
28	176.605150	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8ec5, seq=4/1024, ttl=64 (request in 27)
29	177.606040	10.0.0.1	10.0.0.3	ICMP	98	Echo (ping) request id=0x8fc5, seq=5/1280, ttl=64 (reply in 30)
30	177.606387	10.0.0.3	10.0.0.1	ICMP	98	Echo (ping) reply id=0x8fc5, seq=5/1280, ttl=64 (request in 29)
31	209.396015	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.0.2? Tell 10.0.0.1 [ETHERNET FRAME CHECK SEQUENCE IN…

▶ Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface 0
▶ Ethernet II, Src: Private_66:68:02 (00:50:79:66:68:02), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▶ Address Resolution Protocol (request/gratuitous ARP)

0000 ff ff ff ff ff ff 00 50 79 66 68 02 08 06 00 01   ·······P yfh····

Ready to load or capture   Packets: 42 · Displayed: 42 (100.0%)   Profile: Default

PC1 Terminal(Next Page)

arp -a

MAC addresses are mentioned above for all the 4 PC's.