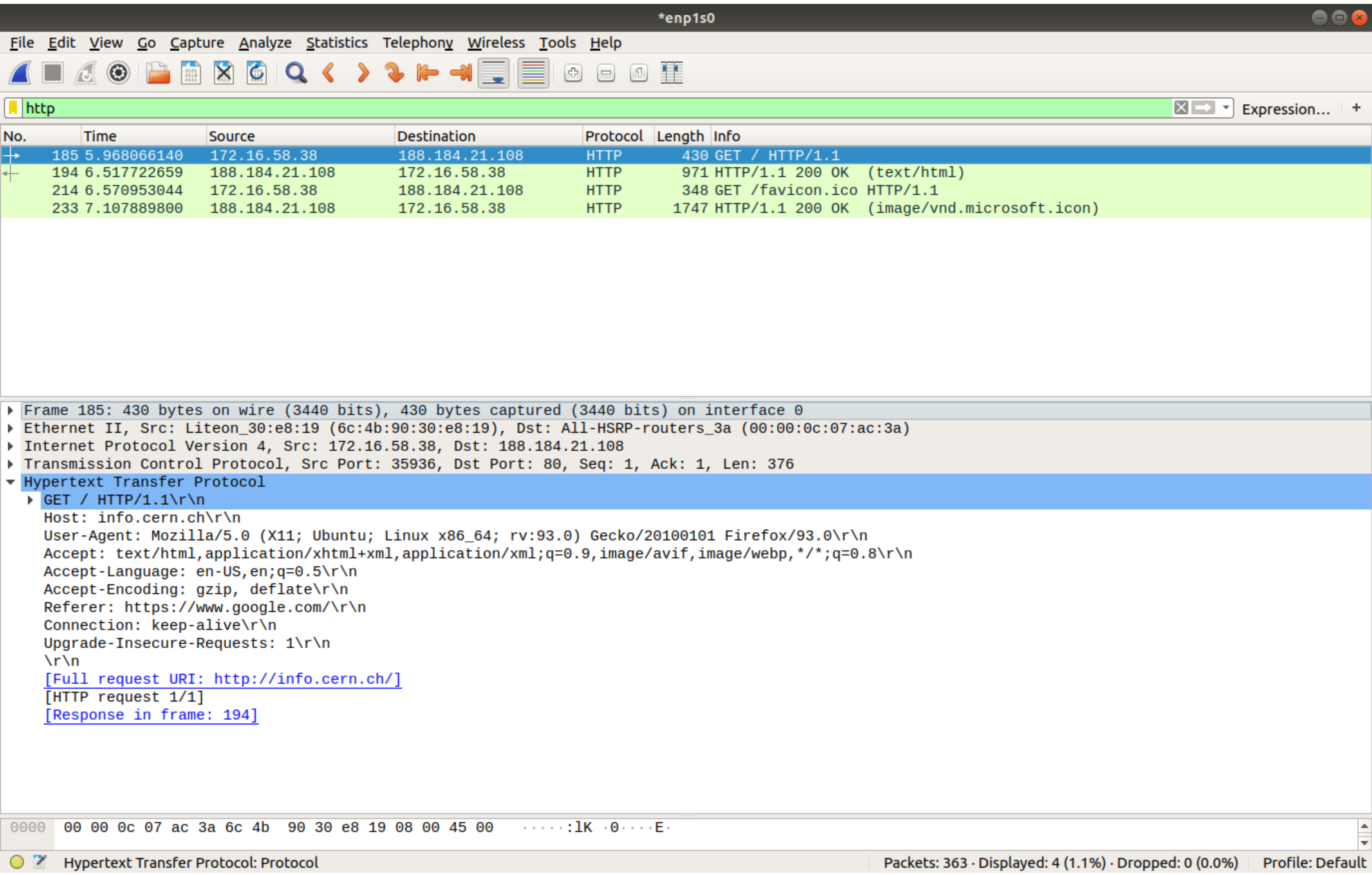
190905104
Lab 3

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.

Retrieving http://info.cern.ch/


*enp1s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

```
http
```

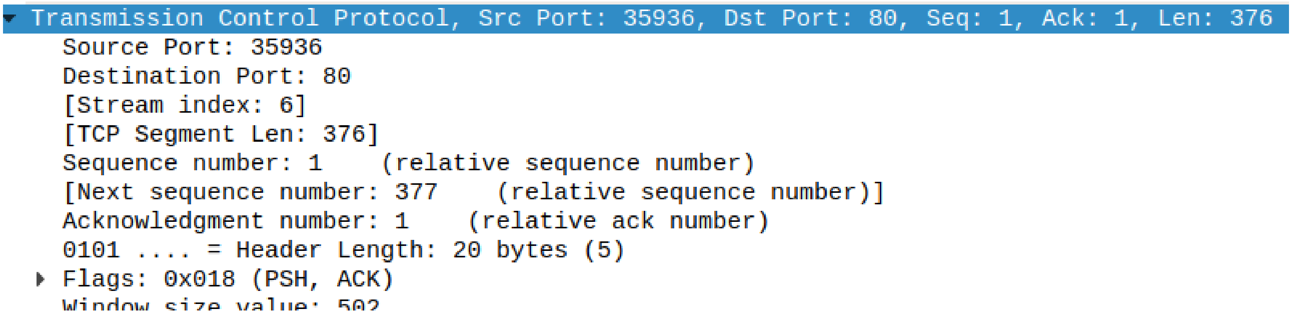
```
No.   Time         Source          Destination     Protocol Length Info
  185 5.968066140  172.16.58.38    188.184.21.108  HTTP      430 GET / HTTP/1.1
  194 6.517722659  188.184.21.108  172.16.58.38    HTTP      971 HTTP/1.1 200 OK  (text/html)
  214 6.570953044  172.16.58.38    188.184.21.108  HTTP      348 GET /favicon.ico HTTP/1.1
  233 7.107889800  188.184.21.108  172.16.58.38    HTTP     1747 HTTP/1.1 200 OK  (image/vnd.microsoft.icon)
```

```
▸ Frame 185: 430 bytes on wire (3440 bits), 430 bytes captured (3440 bits) on interface 0
▸ Ethernet II, Src: Liteon_30:e8:19 (6c:4b:90:30:e8:19), Dst: All-HSRP-routers_3a (00:00:0c:07:ac:3a)
▸ Internet Protocol Version 4, Src: 172.16.58.38, Dst: 188.184.21.108
▸ Transmission Control Protocol, Src Port: 35936, Dst Port: 80, Seq: 1, Ack: 1, Len: 376
▾ Hypertext Transfer Protocol
  ▸ GET / HTTP/1.1\r\n
    Host: info.cern.ch\r\n
    User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:93.0) Gecko/20100101 Firefox/93.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: https://www.google.com/\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://info.cern.ch/]
    [HTTP request 1/1]
    [Response in frame: 194]
```

```
0000  00 00 0c 07 ac 3a 6c 4b  90 30 e8 19 08 00 45 00   ·····:lK ·0····E·
```

```
Hypertext Transfer Protocol: Protocol          Packets: 363 · Displayed: 4 (1.1%) · Dropped: 0 (0.0%)    Profile: Default
```

From the screenshot we can see that the IP address of the source is 172.16.58.38 and the IP address of destination is 188.184.21.108 for the GET request. The GET request is sent when we press enter after typing the URL. The Connection: keep-alive indicates a persistant connection, is an instruction that allows a single TCP connection to remain open for multiple HTTP requests/responses.

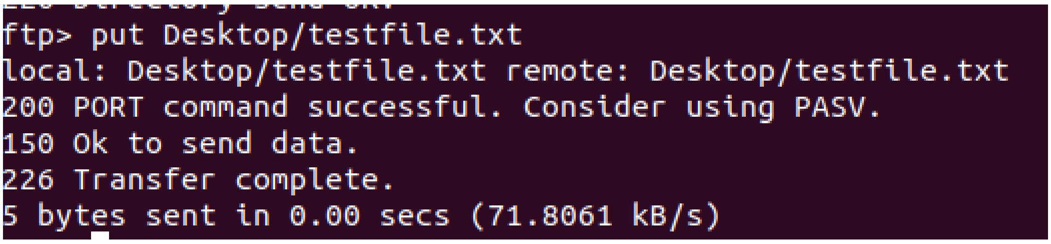


```
▾ Transmission Control Protocol, Src Port: 35936, Dst Port: 80, Seq: 1, Ack: 1, Len: 376
    Source Port: 35936
    Destination Port: 80
    [Stream index: 6]
    [TCP Segment Len: 376]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 377    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 502
```

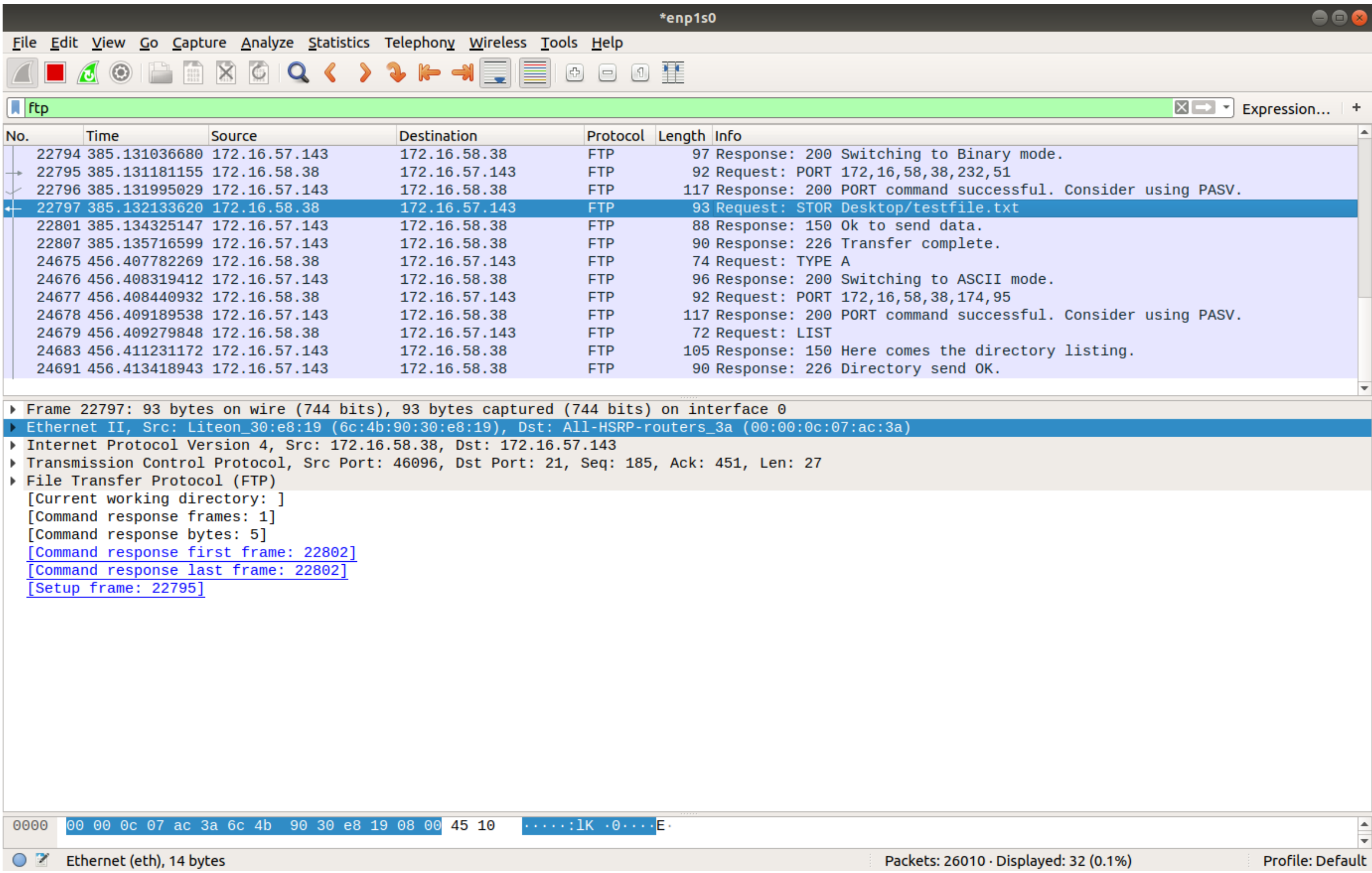
We can see the source port(35936) and destination port(80).

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.

To put a file to the remote server.



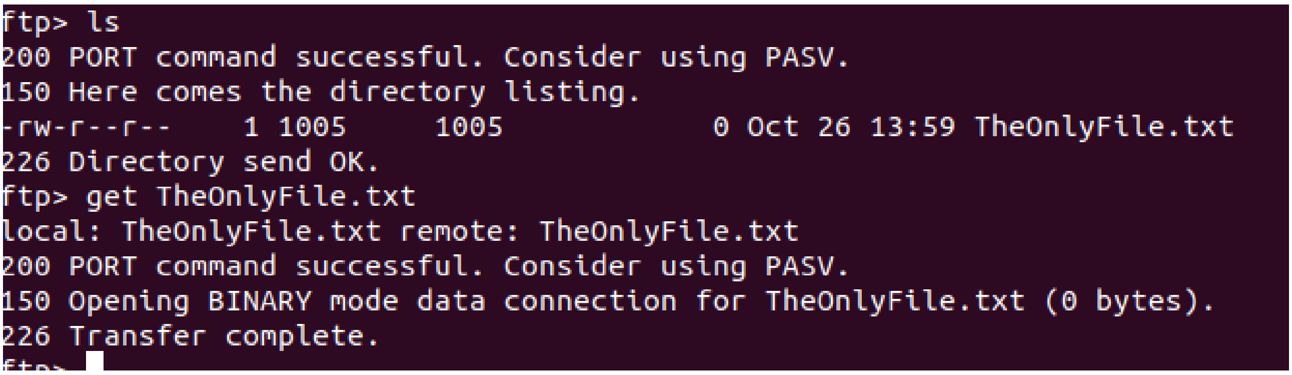
```
ftp> put Desktop/testfile.txt
local: Desktop/testfile.txt remote: Desktop/testfile.txt
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
5 bytes sent in 0.00 secs (71.8061 kB/s)
```



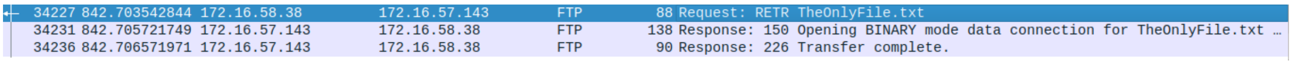
*enp1s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ftp
Expression…
+

```
No.   Time           Source         Destination    Protocol Length Info
22794 385.131036680 172.16.57.143 172.16.58.38   FTP        97 Response: 200 Switching to Binary mode.
22795 385.131181155 172.16.58.38  172.16.57.143  FTP        92 Request: PORT 172,16,58,38,232,51
22796 385.131995029 172.16.57.143 172.16.58.38   FTP       117 Response: 200 PORT command successful. Consider using PASV.
22797 385.132133620 172.16.58.38  172.16.57.143  FTP        93 Request: STOR Desktop/testfile.txt
22801 385.134325147 172.16.57.143 172.16.58.38   FTP        88 Response: 150 Ok to send data.
22807 385.135716599 172.16.57.143 172.16.58.38   FTP        90 Response: 226 Transfer complete.
24675 456.407782269 172.16.58.38  172.16.57.143  FTP        74 Request: TYPE A
24676 456.408319412 172.16.57.143 172.16.58.38   FTP        96 Response: 200 Switching to ASCII mode.
24677 456.408440932 172.16.58.38  172.16.57.143  FTP        92 Request: PORT 172,16,58,38,174,95
24678 456.409189538 172.16.57.143 172.16.58.38   FTP       117 Response: 200 PORT command successful. Consider using PASV.
24679 456.409279848 172.16.58.38  172.16.57.143  FTP        72 Request: LIST
24683 456.411231172 172.16.57.143 172.16.58.38   FTP       105 Response: 150 Here comes the directory listing.
24691 456.413418943 172.16.57.143 172.16.58.38   FTP        90 Response: 226 Directory send OK.

▸ Frame 22797: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface 0
▸ Ethernet II, Src: Liteon_30:e8:19 (6c:4b:90:30:e8:19), Dst: All-HSRP-routers_3a (00:00:0c:07:ac:3a)
▸ Internet Protocol Version 4, Src: 172.16.58.38, Dst: 172.16.57.143
▸ Transmission Control Protocol, Src Port: 46096, Dst Port: 21, Seq: 185, Ack: 451, Len: 27
▸ File Transfer Protocol (FTP)
  [Current working directory: ]
  [Command response frames: 1]
  [Command response bytes: 5]
  [Command response first frame: 22802]
  [Command response last frame: 22802]
  [Setup frame: 22795]

0000  00 00 0c 07 ac 3a 6c 4b  90 30 e8 19 08 00 45 10   ·····:lK ·0····E·

Ethernet (eth), 14 bytes          Packets: 26010 · Displayed: 32 (0.1%)          Profile: Default
```

TCP uses dest port 21 to transfer data and 20 to establish connection.
Using GET to get a file from the server



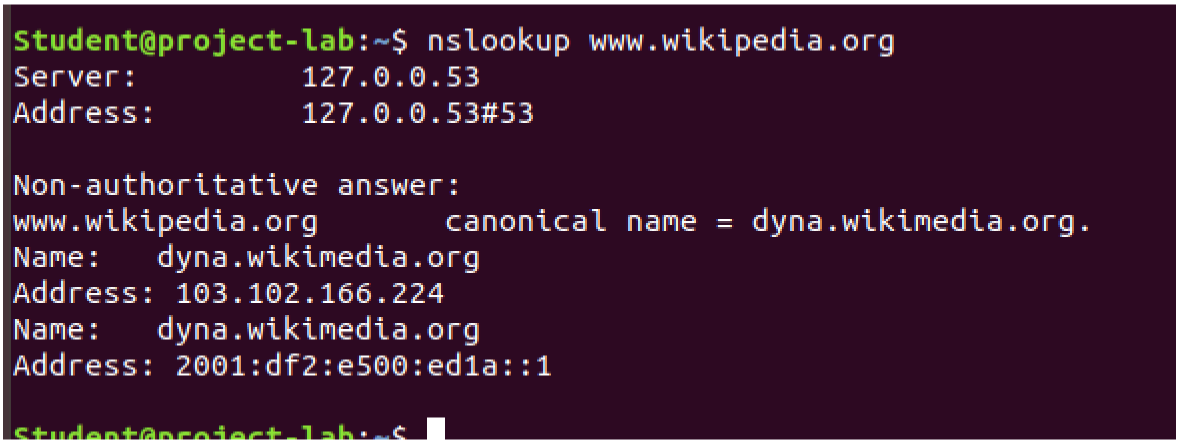
```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 1005     1005            0 Oct 26 13:59 TheOnlyFile.txt
226 Directory send OK.
ftp> get TheOnlyFile.txt
local: TheOnlyFile.txt remote: TheOnlyFile.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for TheOnlyFile.txt (0 bytes).
226 Transfer complete.
ftp>
```



```
34227 842.703542844 172.16.58.38    172.16.57.143   FTP    88 Request: RETR TheOnlyFile.txt
34231 842.705721749 172.16.57.143   172.16.58.38    FTP   138 Response: 150 Opening BINARY mode data connection for TheOnlyFile.txt …
34236 842.706571971 172.16.57.143   172.16.58.38    FTP    90 Response: 226 Transfer complete.
```

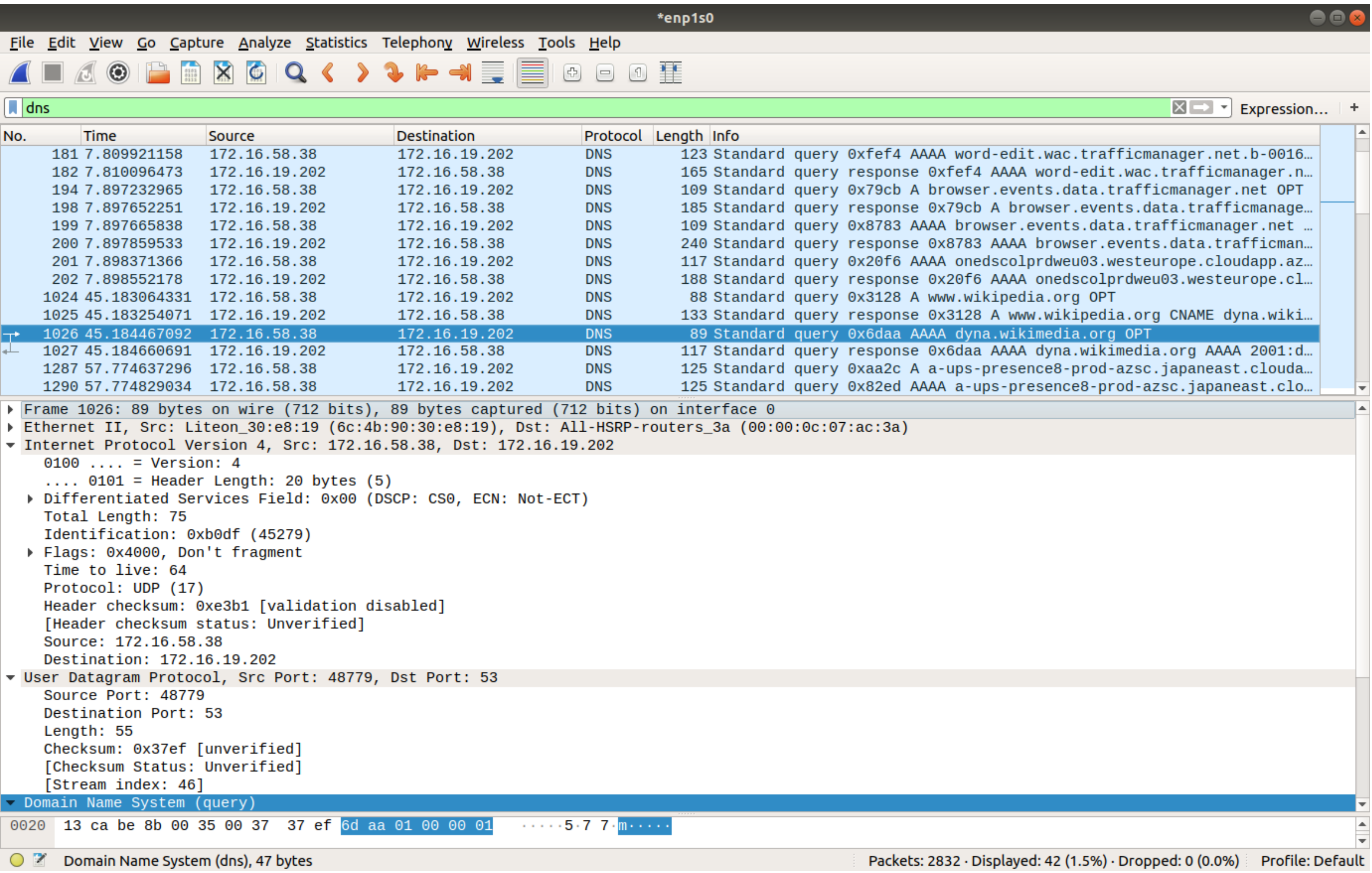
FTP is an insecure transfer protocol since the password is in plaintext and no excryption is used.

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.]



```
Student@project-lab:~$ nslookup www.wikipedia.org
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
www.wikipedia.org       canonical name = dyna.wikimedia.org.
Name:   dyna.wikimedia.org
Address: 103.102.166.224
Name:   dyna.wikimedia.org
Address: 2001:df2:e500:ed1a::1

Student@project-lab:~$
```

DNS protocol when a request for www.wikipedia.org is made. As seen in the image below, DNS uses the UDP protocol in the transport layer. The source port for the UDP protocol is 48779 and the destination port is 53. The request is sent from source address 172.16.58.38 to destination address 172.16.19.202.


*enp1s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression…
+

```
dns

No.  Time          Source         Destination    Protocol Length Info
 181 7.809921158   172.16.58.38   172.16.19.202  DNS      123 Standard query 0xfef4 AAAA word-edit.wac.trafficmanager.net.b-0016…
 182 7.810096473   172.16.19.202  172.16.58.38   DNS      165 Standard query response 0xfef4 AAAA word-edit.wac.trafficmanager.n…
 194 7.897232965   172.16.58.38   172.16.19.202  DNS      109 Standard query 0x79cb A browser.events.data.trafficmanager.net OPT
 198 7.897652251   172.16.19.202  172.16.58.38   DNS      185 Standard query response 0x79cb A browser.events.data.trafficmanage…
 199 7.897665838   172.16.58.38   172.16.19.202  DNS      109 Standard query 0x8783 AAAA browser.events.data.trafficmanager.net …
 200 7.897859533   172.16.19.202  172.16.58.38   DNS      240 Standard query response 0x8783 AAAA browser.events.data.trafficman…
 201 7.898371366   172.16.58.38   172.16.19.202  DNS      117 Standard query 0x20f6 AAAA onedscolprdweu03.westeurope.cloudapp.az…
 202 7.898552178   172.16.19.202  172.16.58.38   DNS      188 Standard query response 0x20f6 AAAA onedscolprdweu03.westeurope.cl…
1024 45.183064331  172.16.58.38   172.16.19.202  DNS       88 Standard query 0x3128 A www.wikipedia.org OPT
1025 45.183254071  172.16.19.202  172.16.58.38   DNS      133 Standard query response 0x3128 A www.wikipedia.org CNAME dyna.wiki…
1026 45.184467092  172.16.58.38   172.16.19.202  DNS       89 Standard query 0x6daa AAAA dyna.wikimedia.org OPT
1027 45.184660691  172.16.19.202  172.16.58.38   DNS      117 Standard query response 0x6daa AAAA dyna.wikimedia.org AAAA 2001:d…
1287 57.774637296  172.16.58.38   172.16.19.202  DNS      125 Standard query 0xaa2c A a-ups-presence8-prod-azsc.japaneast.clouda…
1290 57.774829034  172.16.58.38   172.16.19.202  DNS      125 Standard query 0x82ed AAAA a-ups-presence8-prod-azsc.japaneast.clo…

▸ Frame 1026: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface 0
▸ Ethernet II, Src: Liteon_30:e8:19 (6c:4b:90:30:e8:19), Dst: All-HSRP-routers_3a (00:00:0c:07:ac:3a)
▾ Internet Protocol Version 4, Src: 172.16.58.38, Dst: 172.16.19.202
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 75
    Identification: 0xb0df (45279)
  ▸ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0xe3b1 [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.16.58.38
    Destination: 172.16.19.202
▾ User Datagram Protocol, Src Port: 48779, Dst Port: 53
    Source Port: 48779
    Destination Port: 53
    Length: 55
    Checksum: 0x37ef [unverified]
    [Checksum Status: Unverified]
    [Stream index: 46]
▾ Domain Name System (query)

0020  13 ca be 8b 00 35 00 37  37 ef 6d aa 01 00 00 01   ·····5·7 7·m·····

Domain Name System (dns), 47 bytes          Packets: 2832 · Displayed: 42 (1.5%) · Dropped: 0 (0.0%)    Profile: Default
```

4) Design network configuration shown in Figure 4.1 for all parts. Connect all four VMs to a single Ethernet segment via a single hub as shown in Figure 4.1. Configure the IP addresses for the PCs as shown in Table 4.1.

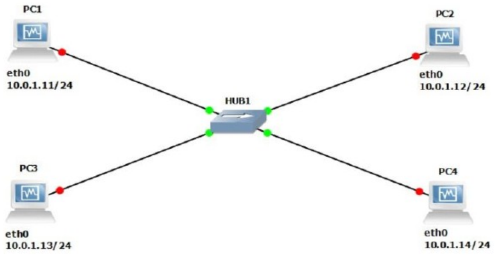
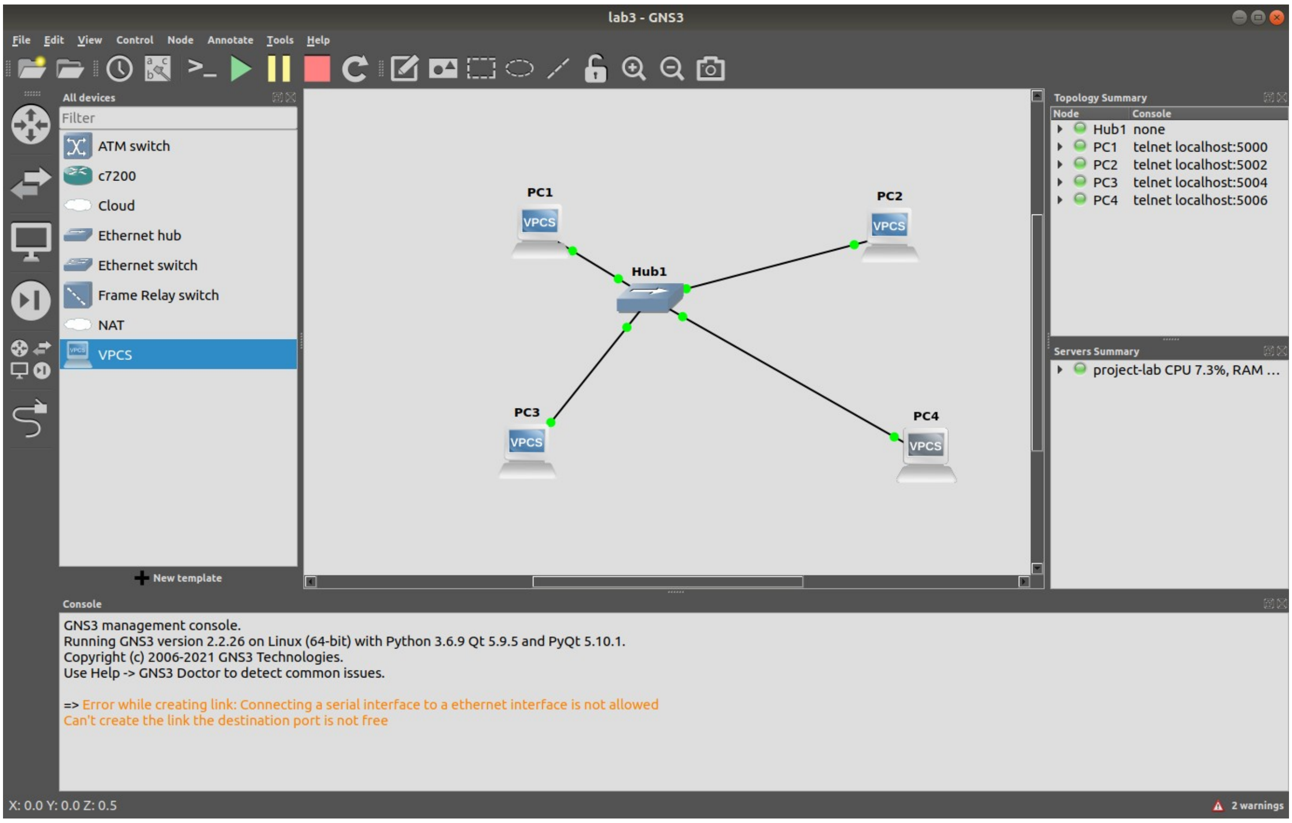


Figure 4.1: Network Design

VMS	IP Addresses of Ethernet Interface eth0
PC1	10.0.1.11 / 24
PC2	10.0.1.12 / 24
PC3	10.0.1.13 / 24
PC4	10.0.1.14 / 24

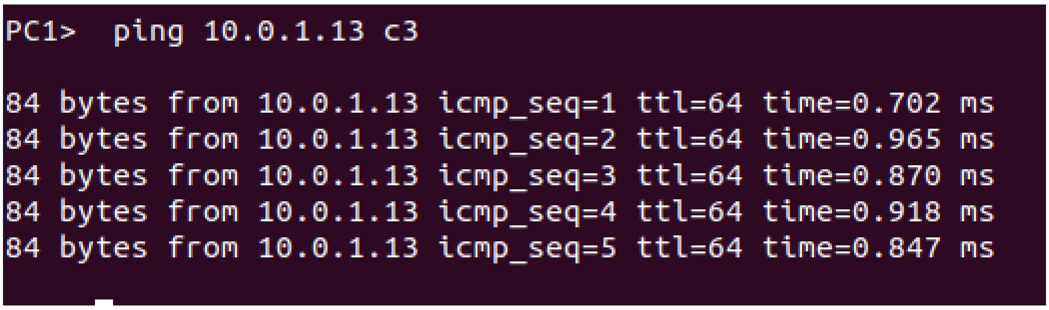


```
Hub1  none
PC1   telnet localhost:5000
PC2   telnet localhost:5002
PC3   telnet localhost:5004
PC4   telnet localhost:5006
```

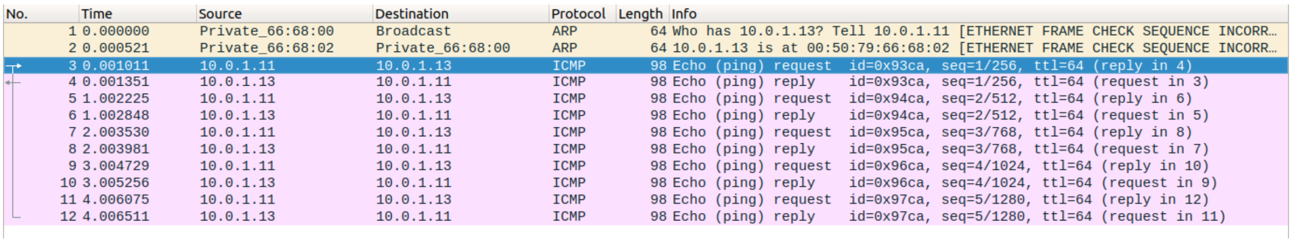
```
project-lab CPU 7.3%, RAM ...
```

```
GNS3 management console.
Running GNS3 version 2.2.26 on Linux (64-bit) with Python 3.6.9 Qt 5.9.5 and PyQt 5.10.1.
Copyright (c) 2006-2021 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=> Error while creating link: Connecting a serial interface to a ethernet interface is not allowed
Can't create the link the destination port is not free
```

Pinging PC3 from PC1

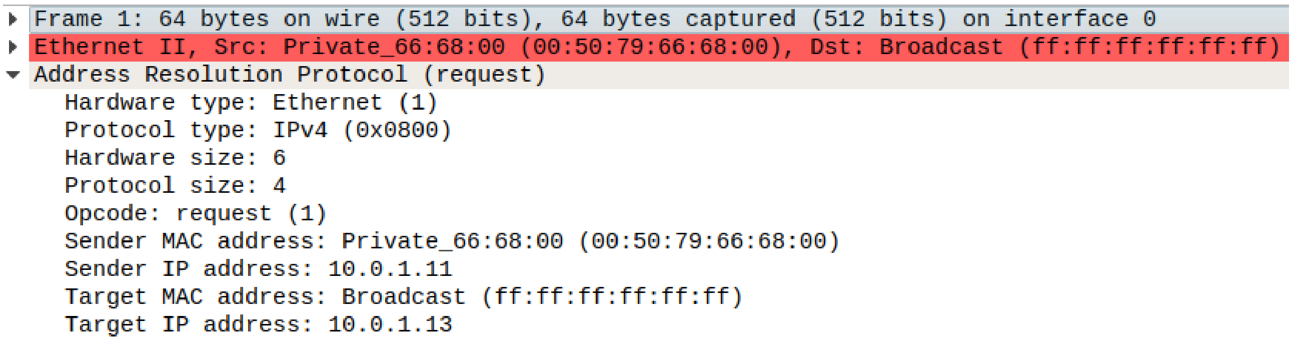


```
PC1>  ping 10.0.1.13 c3

84 bytes from 10.0.1.13 icmp_seq=1 ttl=64 time=0.702 ms
84 bytes from 10.0.1.13 icmp_seq=2 ttl=64 time=0.965 ms
84 bytes from 10.0.1.13 icmp_seq=3 ttl=64 time=0.870 ms
84 bytes from 10.0.1.13 icmp_seq=4 ttl=64 time=0.918 ms
84 bytes from 10.0.1.13 icmp_seq=5 ttl=64 time=0.847 ms
```



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Private_66:68:00	Broadcast	ARP	64	Who has 10.0.1.13? Tell 10.0.1.11 [ETHERNET FRAME CHECK SEQUENCE INCORR…
2	0.000521	Private_66:68:02	Private_66:68:00	ARP	64	10.0.1.13 is at 00:50:79:66:68:02 [ETHERNET FRAME CHECK SEQUENCE INCORR…
3	0.001011	10.0.1.11	10.0.1.13	ICMP	98	Echo (ping) request id=0x93ca, seq=1/256, ttl=64 (reply in 4)
4	0.001351	10.0.1.13	10.0.1.11	ICMP	98	Echo (ping) reply id=0x93ca, seq=1/256, ttl=64 (request in 3)
5	1.002225	10.0.1.11	10.0.1.13	ICMP	98	Echo (ping) request id=0x94ca, seq=2/512, ttl=64 (reply in 6)
6	1.002848	10.0.1.13	10.0.1.11	ICMP	98	Echo (ping) reply id=0x94ca, seq=2/512, ttl=64 (request in 5)
7	2.003530	10.0.1.11	10.0.1.13	ICMP	98	Echo (ping) request id=0x95ca, seq=3/768, ttl=64 (reply in 8)
8	2.003981	10.0.1.13	10.0.1.11	ICMP	98	Echo (ping) reply id=0x95ca, seq=3/768, ttl=64 (request in 7)
9	3.004729	10.0.1.11	10.0.1.13	ICMP	98	Echo (ping) request id=0x96ca, seq=4/1024, ttl=64 (reply in 10)
10	3.005256	10.0.1.13	10.0.1.11	ICMP	98	Echo (ping) reply id=0x96ca, seq=4/1024, ttl=64 (request in 9)
11	4.006075	10.0.1.11	10.0.1.13	ICMP	98	Echo (ping) request id=0x97ca, seq=5/1280, ttl=64 (reply in 12)
12	4.006511	10.0.1.13	10.0.1.11	ICMP	98	Echo (ping) reply id=0x97ca, seq=5/1280, ttl=64 (request in 11)

Total 10 calls are made between PC2 and PC1, 2 for each ping i.e one is request and the other is the reply.



```
▸ 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)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Private_66:68:00 (00:50:79:66:68:00)
    Sender IP address: 10.0.1.11
    Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
    Target IP address: 10.0.1.13
```