

Gasser Ahmed
gasser18@vt.edu
 ECE 5484, Project 3

Section 1 – Objectives:

Reinforce the understanding of network protocols and the TCP/IP protocol suite by using Wireshark network protocol analyzer.

By following the steps below:

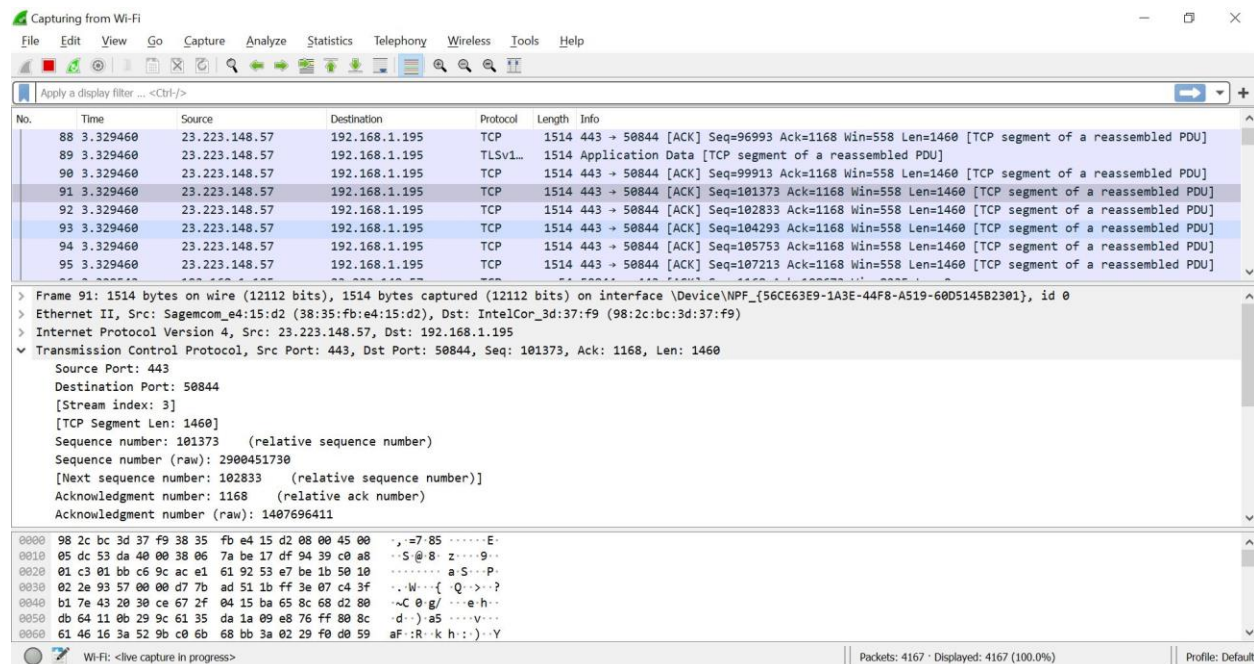
1. Install and become familiar with the basic operation of Wireshark.
2. Capture and analyze a simple HTTP transaction.

Section 2 – Wireshark Installation:

I was able to install Wireshark successfully. The installation process was very straight forward, and I did not face any problems or issues. Having Npcap waiting for my input to move forward with the setup while I was on the main Wireshark setup window was probably the trickiest part of the setup.

Section 3 – Familiarization:

The familiarization process was straightforward and I didn't have any problems encountered during that phase.



Section 4 – Protocol Analysis (Screenshots and Answers):

The image shows a Wireshark packet capture analysis of an HTTP GET request. The top pane displays a list of packets, with packet 7 selected. The middle pane shows the packet details for the selected packet, and the bottom pane shows the raw packet data in hexadecimal and ASCII.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.40.14	70.40.220.247	TCP	78	64145 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=259357741 TSecr=0 SACK_PERM=0
2	0.000705	192.168.40.14	70.40.220.247	TCP	78	64146 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=259357749 TSecr=0 SACK_PERM=0
3	0.106996	70.40.220.247	192.168.40.14	TCP	74	80 → 64145 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2356942350
4	0.107085	192.168.40.14	70.40.220.247	TCP	66	64145 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=259357846 TSecr=2356942350
5	0.109681	70.40.220.247	192.168.40.14	TCP	74	80 → 64146 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2356942350
6	0.109778	192.168.40.14	70.40.220.247	TCP	66	64146 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=259357848 TSecr=2356942350
7	0.110075	192.168.40.14	70.40.220.247	HTTP	414	GET /egg.txt HTTP/1.1
8	0.213787	70.40.220.247	192.168.40.14	TCP	66	80 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=0 TSval=2356942361 TSecr=259357848

Packet Details:

- Frame 7: 414 bytes on wire (3312 bits), 414 bytes captured (3312 bits) on interface en0, id 0
- Ethernet II, Src: Apple_97:25:8a (6c:4d:08:97:25:8a), Dst: Apple_58:61:7d (b8:bd:12:58:61:7d)
- Internet Protocol Version 4, Src: 192.168.40.14, Dst: 70.40.220.247
 - 0100 = Version: 4
 - 0101 = Header Length: 20 bytes (5)
- Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 400
 - Identification: 0x0000 (0)
- Flags: 0x4000, Don't fragment
 - Fragment offset: 0
 - Time to live: 64
 - Protocol: TCP (6)
 - Header checksum: 0x2d92 [validation disabled]

Raw Data:

```

0000 b8 8d 12 58 61 7d 6c 40 08 97 25 8a 08 00 00 00  ...Xa)l@...%..E.
0010 01 90 00 00 40 00 40 06 2d 92 c0 a8 28 0e 46 28  ...@.@....(-F(
0020 dc f7 fa 92 00 50 12 ab 0e 54 b4 4a dd 33 80 18  ...P...T.J.3..
0030 10 15 cb ac 00 00 01 01 00 0a 0f 75 7c 98 8c 7c  ...GET.....u|..|
0040 16 0e 47 45 54 20 2f 65 67 27 74 78 74 20 48  .../e/gg.txt H
0050 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77  TTP/1.1..Host: w
0060 77 77 2e 6f 72 69 6f 6e 73 71 62 72 6f 77 2e 63  ww.orion.sarrow.c

```

1. HTTP Response time for egg.txt: $0.323551 - 0.110075 = 0.213476$ s

Wireshark - Flow - egg_bt_dump.pcapng

Time	192.168.40.14	70.40.220.247	Comment
0.109681	64146 → 64146 [SYN, ACK] Seq=0 Ack=1 Win=20480	80	TCP: 80 → 64146 [SYN, ACK] Seq=0 Ack=1 Win=...
0.109778	64146 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0	80	TCP: 64146 → 80 [ACK] Seq=1 Ack=1 Win=13174...
0.110075	64146 → 80 [GET] /egg.txt HTTP/1.1	80	HTTP: GET /egg.txt HTTP/1.1
0.213787	64146 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=0	80	TCP: 80 → 64146 [ACK] Seq=1 Ack=349 Win=300...
0.222416	64146 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=0	80	TCP: 80 → 64146 [ACK] Seq=1 Ack=349 Win=300...
0.222679	64146 → 64146 [ACK] Seq=1449 Ack=349 Win=20480	80	TCP: 80 → 64146 [ACK] Seq=1449 Ack=349 Win=...
0.222742	64146 → 80 [ACK] Seq=349 Ack=2897 Win=12960	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=2897 Win=...
0.232019	64146 → 80 [ACK] Seq=2897 Ack=349 Win=30080	80	TCP: 80 → 64146 [ACK] Seq=2897 Ack=349 Win=...
0.232106	64146 → 80 [ACK] Seq=349 Ack=4345 Win=11307	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=4345 Win=...
0.232023	64146 → 64146 [ACK] Seq=4345 Ack=349 Win=30080	80	TCP: 80 → 64146 [ACK] Seq=4345 Ack=349 Win=...
0.230840	64146 → 64146 [ACK] Seq=5793 Ack=349 Win=30080	80	TCP: 80 → 64146 [ACK] Seq=5793 Ack=349 Win=...
0.230971	64146 → 80 [ACK] Seq=349 Ack=7241 Win=12960	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=7241 Win=...
0.231180	64146 → 64146 [ACK] Seq=7241 Ack=349 Win=30080	80	TCP: 80 → 64146 [ACK] Seq=7241 Ack=349 Win=...
0.231289	64146 → 80 [ACK] Seq=349 Ack=8689 Win=11307	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=8689 Win=...
0.232077	64146 → 64146 [ACK] Seq=8689 Ack=349 Win=30080	80	TCP: 80 → 64146 [ACK] Seq=8689 Ack=349 Win=...
0.232312	64146 → 64146 [ACK] Seq=10137 Ack=349 Win=3008	80	TCP: 80 → 64146 [ACK] Seq=10137 Ack=349 Win=...
0.232361	64146 → 80 [ACK] Seq=349 Ack=11585 Win=12960	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=11585 Win=...
0.232704	64146 → 64146 [ACK] Seq=11585 Ack=349 Win=3008	80	TCP: 80 → 64146 [ACK] Seq=11585 Ack=349 Win=...
0.232770	64146 → 80 [ACK] Seq=349 Ack=13033 Win=1310	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=13033 Win=...
0.233088	64146 → 64146 [ACK] Seq=13033 Ack=349 Win=3008	80	TCP: 80 → 64146 [ACK] Seq=13033 Ack=349 Win=...
0.322510	64146 → 64146 [ACK] Seq=14481 Ack=349 Win=3008	80	TCP: 80 → 64146 [ACK] Seq=14481 Ack=349 Win=...
0.322593	64146 → 80 [ACK] Seq=349 Ack=15929 Win=12960	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=15929 Win=...
0.323551	64146 → 80 [HTTP/1.1 200 OK (text/plain)]	80	HTTP: HTTP/1.1 200 OK (text/plain)
0.323632	64146 → 80 [ACK] Seq=349 Ack=16976 Win=1300	80	TCP: 64146 → 80 [ACK] Seq=349 Ack=16976 Win=...

Packet 24: TCP: 80 → 64146 [ACK] Seq=13033 Ack=349 Win=30080 Len=1440 TSval=255842361 TSecr=259257948 [TCP segment of a reassembled PDU]

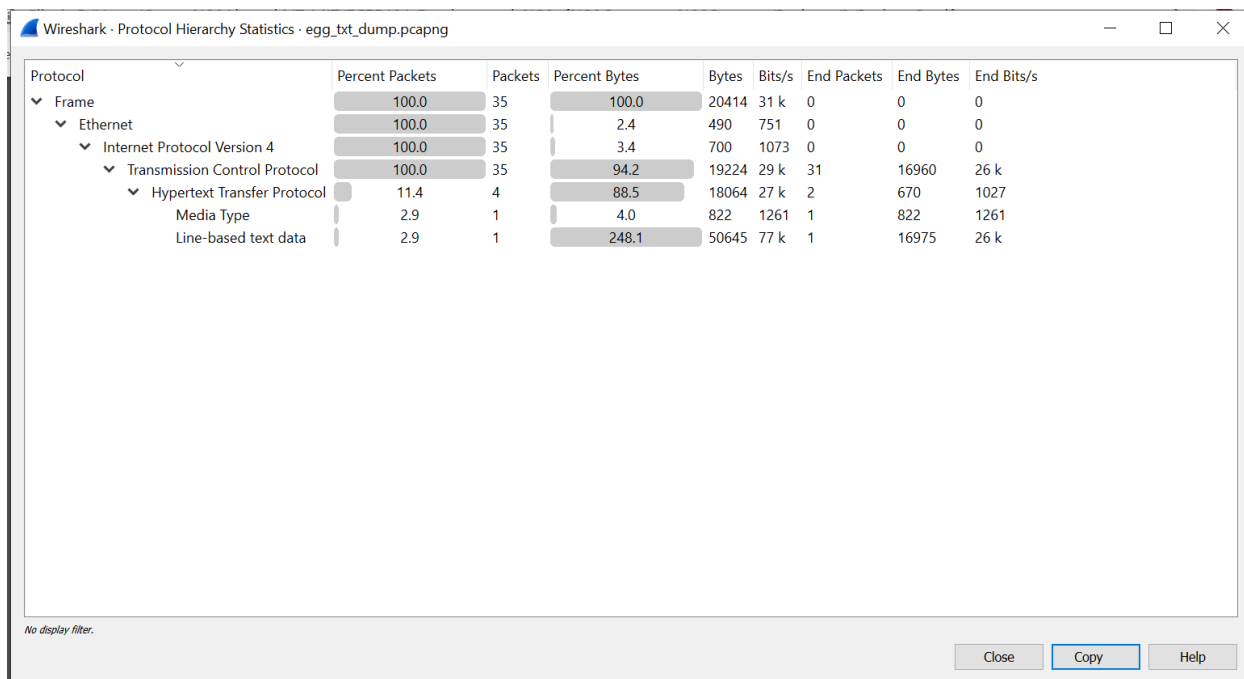
☐ Limit to display filter

Flow type: All Flows

Addresses: Any

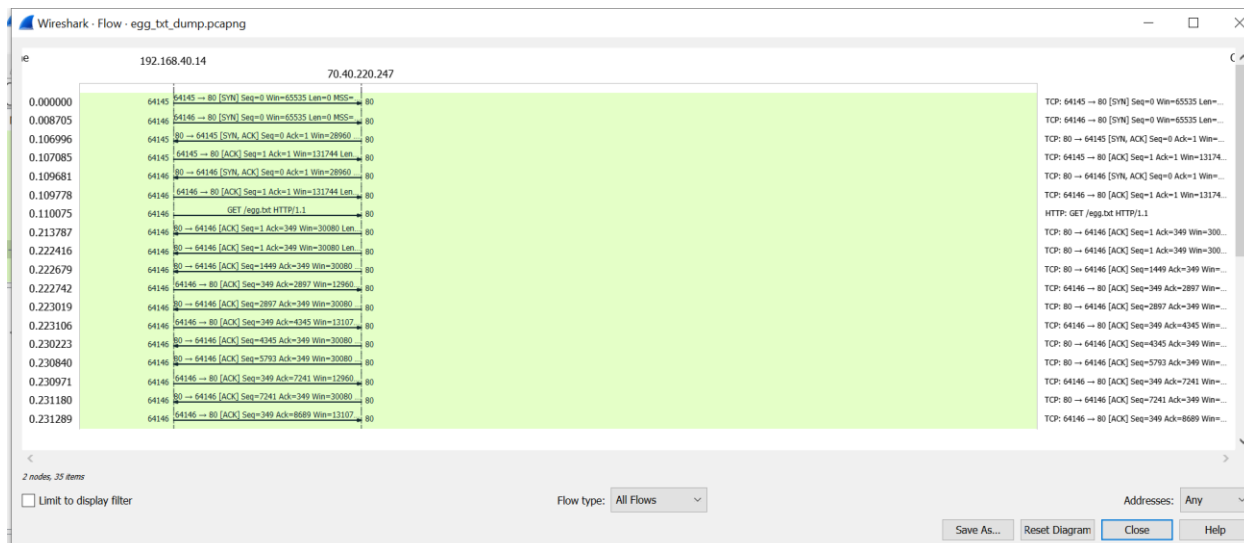
Save As... Reset Diagram Close Help

Protocol Hierarchy:



- Percentage of the captured packets are using TCP: **100%**
- From the captured packets, an example Application Layer protocol that uses TCP: **Hypertext Transfer Protocol (HTTP)**

Flow Graph:



HTTP GET request Questions:

4. How many bytes are in the IP header? 20 bytes

egg_txt_dump.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
7	0.110075	192.168.40.14	70.40.220.247	HTTP	414	GET /egg.txt HTTP/1.1
27	0.323551	70.40.220.247	192.168.40.14	HTTP	1113	HTTP/1.1 200 OK (text/plain)
29	0.332767	192.168.40.14	70.40.220.247	HTTP	388	GET /favicon.ico HTTP/1.1
30	0.436974	70.40.220.247	192.168.40.14	HTTP	485	HTTP/1.1 200 OK (image/vnd.microsoft.icon)

Internet Protocol Version 4, Src: 192.168.40.14, Dst: 70.40.220.247

- 0100 = Version: 4
- 0101 = Header Length: 20 bytes (5)
- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 400
- Identification: 0x0000 (0)
- > Flags: 0x4000, Don't fragment
- Fragment offset: 0
- Time to live: 64
- Protocol: TCP (6)
- Header checksum: 0x2d92 [validation disabled]
- [Header checksum status: Unverified]
- Source: 192.168.40.14

5. How many bytes are in the TCP header? 32 bytes

egg_txt_dump.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http

No.	Time	Source	Destination	Protocol	Length	Info
7	0.110075	192.168.40.14	70.40.220.247	HTTP	414	GET /egg.txt HTTP/1.1
27	0.323551	70.40.220.247	192.168.40.14	HTTP	1113	HTTP/1.1 200 OK (text/plain)
29	0.332767	192.168.40.14	70.40.220.247	HTTP	388	GET /favicon.ico HTTP/1.1
30	0.436974	70.40.220.247	192.168.40.14	HTTP	485	HTTP/1.1 200 OK (image/vnd.microsoft.icon)

Transmission Control Protocol, Src Port: 64146, Dst Port: 80, Seq: 1, Ack: 1, Len: 348

- Source Port: 64146
- Destination Port: 80
- [Stream index: 1]
- [TCP Segment Len: 348]
- Sequence number: 1 (relative sequence number)
- Sequence number (raw): 313200212
- [Next sequence number: 349 (relative sequence number)]
- Acknowledgment number: 1 (relative ack number)
- Acknowledgment number (raw): 3024805171
- 1000 = Header Length: 32 bytes (8)

6. How many bytes are in the HTTP message? **348 bytes**

egg_bt_dump.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
7	0.110875	192.168.40.14	70.40.220.247	HTTP	414	GET /egg.txt HTTP/1.1
8	0.213787	70.40.220.247	192.168.40.14	TCP	66	80 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=0 TSval=2356942361 TSecr=259357848
9	0.222416	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]
10	0.222679	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=1449 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]
11	0.222742	192.168.40.14	70.40.220.247	TCP	66	64146 → 80 [ACK] Seq=349 Ack=2897 Win=129600 Len=0 TSval=259357959 TSecr=2356942361
12	0.223819	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=2897 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]
13	0.223196	192.168.40.14	70.40.220.247	TCP	66	64146 → 80 [ACK] Seq=349 Ack=4345 Win=131072 Len=0 TSval=259357959 TSecr=2356942361
14	0.230223	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=4345 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]

> Ethernet II, Src: Apple_97:25:8a (6c:40:08:97:25:8a), Dst: Apple_58:61:7d (b8:8d:12:58:61:7d)

> Internet Protocol Version 4, Src: 192.168.40.14, Dst: 70.40.220.247

> Transmission Control Protocol, Src Port: 64146, Dst Port: 80, Seq: 1, Ack: 1, Len: 348

> Hypertext Transfer Protocol

> GET /egg.txt HTTP/1.1\r\n

Host: www.orionsarrow.com\r\n

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:63.0) Gecko/20100101 Firefox/63.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

DNT: 1\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

Hypertext Transfer Protocol (http) 348 bytes

Packets: 35 · Displayed: 35 (100.0%) Profile: Default

7. What is the total length, in bytes, of the IP datagram carrying the IP header, the TCP header, and the HTTP message for the GET request? **400 bytes**

(either by summing results of steps 4, 5, and 6 or by directly getting that number from Total Length under “Internet Protocol Version 4”)

egg_bt_dump.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
7	0.110875	192.168.40.14	70.40.220.247	HTTP	414	GET /egg.txt HTTP/1.1
8	0.213787	70.40.220.247	192.168.40.14	TCP	66	80 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=0 TSval=2356942361 TSecr=259357848
9	0.222416	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=1 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]
10	0.222679	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=1449 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]
11	0.222742	192.168.40.14	70.40.220.247	TCP	66	64146 → 80 [ACK] Seq=349 Ack=2897 Win=129600 Len=0 TSval=259357959 TSecr=2356942361
12	0.223819	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=2897 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]
13	0.223196	192.168.40.14	70.40.220.247	TCP	66	64146 → 80 [ACK] Seq=349 Ack=4345 Win=131072 Len=0 TSval=259357959 TSecr=2356942361
14	0.230223	70.40.220.247	192.168.40.14	TCP	1514	80 → 64146 [ACK] Seq=4345 Ack=349 Win=30080 Len=1448 TSval=2356942361 TSecr=259357848 [TCP...]

> Frame 7: 414 bytes on wire (3312 bits), 414 bytes captured (3312 bits) on interface en0, id 0

> Ethernet II, Src: Apple_97:25:8a (6c:40:08:97:25:8a), Dst: Apple_58:61:7d (b8:8d:12:58:61:7d)

> Internet Protocol Version 4, Src: 192.168.40.14, Dst: 70.40.220.247

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 400

Identification: 0x0000 (0)

Flags: 0x4000, Don't fragment

Fragment offset: 0

Time to live: 64

Protocol: TCP (6)

Header checksum: 0x2d92 [validation disabled]

0010 01 90 00 00 40 00 40 06 2d 92 c0 a8 28 0e 46 28

0020 dc f7 fa 92 00 50 12 ab 0e 54 b4 4a dd 33 80 18

0030 10 15 cb ac 00 00 01 01 08 0a 0f 75 7c 98 8c 7c

0040 16 0e 47 45 54 20 2f 65 67 67 2e 74 78 74 20 48

0050 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77

0060 77 77 2e 6f 72 69 6f 6e 73 61 72 72 6f 77 2e 63

0070 6f 6d 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20

0080 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 4d 61 63

0090 69 6e 74 6f 73 68 3b 20 49 6e 74 65 6c 20 4d 61

00a0 63 20 4f 53 20 58 20 31 30 2e 31 33 3b 20 72 78

00b0 63 20 4f 53 20 58 20 31 30 2e 31 33 3b 20 72 78

Total Length (ip.len), 2 bytes

Packets: 35 · Displayed: 35 (100.0%) Profile: Default

Section 5 – Conclusions:

After going through the installation, familiarization, and protocol analysis phases, I was able to become more familiar with Wireshark and have a better understanding of what it does and how it works.

However, the protocol analysis part was the trickiest part especially with questions 6 and 7 as it took me a long time to figure out how to get the number of bytes in the HTTP message and total length of the IP datagram.

In general, the project was very helpful in understanding Wireshark and having a deeper look into how network protocols work. Lastly, the approximate number of hours I devoted to the project was about 5-8 hours.
