

CS315: Lab Assignment 2

B Siddharth Prabhu

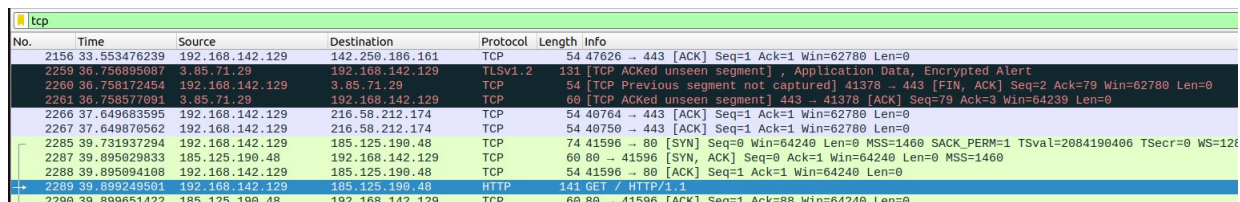
200010003@iitdh.ac.in

10 January 2023

1 Answers for Part 1: Wireshark Basics

(1) If a packet is highlighted by black, what does it mean for the packet?

By default, black highlighting of packets indicates TCP packets with problems. In the below screenshot, we observe that some such problems are 'TCP ACKed unseen segment' (This acknowledges data that wasn't captured) and 'TCP Previous segment not captured' (This means that Wireshark is seeing an acknowledgment for a packet that it hasn't captured).



The screenshot shows a Wireshark packet capture with a filter set to 'tcp'. The packet list pane shows several packets. Packets 2259, 2260, 2261, 2266, 2267, 2285, 2287, 2288, and 2289 are highlighted in black. The packet details pane shows the selected packet (2289) is an HTTP GET request. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
2156	33.553476239	192.168.142.129	142.250.186.161	TCP	54	47626 → 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0
2259	36.756895087	3.85.71.29	192.168.142.129	TLSv1.2	131	[TCP ACKed unseen segment] , Application Data, Encrypted Alert
2260	36.758172454	192.168.142.129	3.85.71.29	TCP	54	[TCP Previous segment not captured] 41378 → 443 [FIN, ACK] Seq=2 Ack=79 Win=62780 Len=0
2261	36.759577091	3.85.71.29	192.168.142.129	TCP	60	[TCP ACKed unseen segment] 443 → 41378 [ACK] Seq=79 Ack=3 Win=64239 Len=0
2266	37.649683595	192.168.142.129	216.58.212.174	TCP	54	40764 → 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0
2267	37.649870562	192.168.142.129	216.58.212.174	TCP	54	40750 → 443 [ACK] Seq=1 Ack=1 Win=62780 Len=0
2285	39.731937294	192.168.142.129	185.125.190.48	TCP	74	41596 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2084190406 TSecr=0 WS=128
2287	39.895029833	185.125.190.48	192.168.142.129	TCP	60	80 → 41596 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
2288	39.895094108	192.168.142.129	185.125.190.48	TCP	54	41596 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
2289	39.899249501	192.168.142.129	185.125.190.48	HTTP	141	GET / HTTP/1.1
2290	39.899651422	185.125.190.48	192.168.142.129	TCP	60	80 → 41596 [ACK] Seq=1 Ack=88 Win=64240 Len=0

Figure 1: Black highlighting

(2) What is the filter command for listing all outgoing HTTP traffic?

Outgoing HTTP traffic consists of HTTP requests, that may be GET or POST. We would like to view all of them, so the filter command is the following:

http.request

(3) Why does DNS use Follow UDP Stream while HTTP use Follow TCP Stream?

UDP (User Datagram Protocol) is a connection-less protocol used to send smaller segments of data on a network, while TCP (Transmission Control Protocol) is a connection-oriented protocol used to send data packets along a network connection.

DNS requests are tiny, and fit in UDP segments easily. Also, we want Domain Name Resolution to occur quickly, so using UDP stream would be a better choice for this purpose. Hence, the main reason for this is performance.

HTTP uses the more reliable TCP stream, since the files, images, and other data that we get from the remote host must not be dropped along the way. Although HTTP could technically use UDP, if a UDP packet containing the first part of a web page is lost, then it is not retransmitted. Then, the application layer would have to handle this. To avoid such overhead burdens, it is better to use TCP instead. Hence, the main reason for this is reliability.

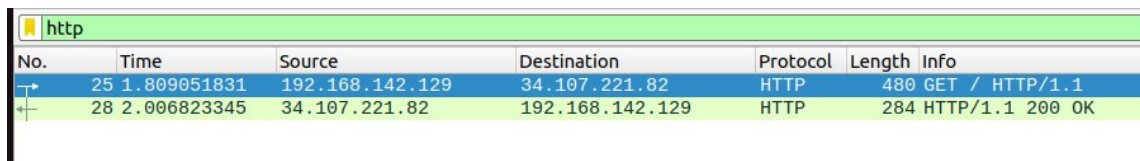
2 Answers for Task 2: Wireshark for Packet Capture and Analysis

(1) List the different protocols that appear in the protocol column in the unfiltered packet-listing window in wireshark GUI?

- TCP : Transmission Control Protocol
- HTTP : HyperText Transfer Protocol
- ARP : Address Resolution Protocol
- QUIC : Quick UDP Internet Connections
- TLSv1.2 : Transport Layer Security (Version 1.2)
- ICMPv6 : Internet Control Message Protocol for IPv6
- NTP : Network Time Protocol
- DNS : Domain Network System (a.k.a. Domain Name System)
- SSDP : Simple Service Discovery Protocol

(2) How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received for the web page you visited in your web browser?

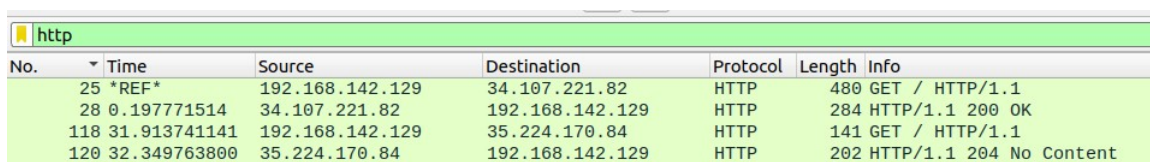
This can be easily obtained by clicking on the HTTP GET record and setting time reference, or by manually subtracting time values. The obtained time difference in this case is 0.197771514 seconds.



The image shows a Wireshark packet capture window with the filter 'http'. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
25	1.809051831	192.168.142.129	34.107.221.82	HTTP	480	GET / HTTP/1.1
28	2.006823345	34.107.221.82	192.168.142.129	HTTP	284	HTTP/1.1 200 OK

Figure 2: Without Setting Time Reference



The image shows the same Wireshark packet capture window, but with the time reference set to the first packet (No. 25). The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
25	*REF*	192.168.142.129	34.107.221.82	HTTP	480	GET / HTTP/1.1
28	0.197771514	34.107.221.82	192.168.142.129	HTTP	284	HTTP/1.1 200 OK
118	31.913741141	192.168.142.129	35.224.170.84	HTTP	141	GET / HTTP/1.1
120	32.349763800	35.224.170.84	192.168.142.129	HTTP	202	HTTP/1.1 204 No Content

Figure 3: With Setting Time Reference

(3) What is the Internet (IP) address of the URL you visited and what is the Internet address of your computer?

IP address of URL visited: 34.107.221.82
IP address of my computer: 192.168.142.129
Note that this is being run on an Ubuntu Virtual Machine.

(4) Print the two HTTP messages displayed in wireshark GUI after you had visited the above URL through your web browser. To do so, select Print from the Wireshark File command menu, and select “Selected Packet Only” and then click Print.

```

No.      Time            Source                Destination            Protocol Length Info
 25      *REF*          192.168.142.129      34.107.221.82         HTTP      480      GET / HTTP/1.1
Frame 25: 480 bytes on wire (3840 bits), 480 bytes captured (3840 bits) on interface ens33, id 0
Ethernet II, Src: VMware_db:81:86 (00:0c:29:db:81:86), Dst: VMware_e3:9c:e6 (00:50:56:e3:9c:e6)
Internet Protocol Version 4, Src: 192.168.142.129, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 51022, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
Hypertext Transfer Protocol

No.      Time            Source                Destination            Protocol Length Info
 28      0.197771514    34.107.221.82        192.168.142.129      HTTP      284      HTTP/1.1 200 OK
Frame 28: 284 bytes on wire (2272 bits), 284 bytes captured (2272 bits) on interface ens33, id 0
Ethernet II, Src: VMware_e3:9c:e6 (00:50:56:e3:9c:e6), Dst: VMware_db:81:86 (00:0c:29:db:81:86)
Internet Protocol Version 4, Src: 34.107.221.82, Dst: 192.168.142.129
Transmission Control Protocol, Src Port: 80, Dst Port: 51022, Seq: 1, Ack: 427, Len: 230
Hypertext Transfer Protocol
Data (8 bytes)
0000  73 75 63 63 65 73 73 0a                                success.

```

Figure 4: HTTP messages

(5) Execute the above steps on Google Chrome, Safari or any other browsers also, check whether you will be able to see http protocol. Write down your analysis with screenshots.

So far, Google Chrome was used. With Mozilla Firefox, the following is obtained:

http						
No.	Time	Source	Destination	Protocol	Length	Info
43	0.818065682	93.184.220.29	192.168.142.129	OCSP	793	Response
62	1.230380591	192.168.142.129	35.224.170.84	HTTP	141	GET / HTTP/1.1
74	1.594143163	35.224.170.84	192.168.142.129	HTTP	202	HTTP/1.1 204 No Content
106	2.558048196	192.168.142.129	2.19.126.223	OCSP	477	Request
108	2.559016896	192.168.142.129	2.19.126.223	OCSP	477	Request
110	2.762616475	2.19.126.223	192.168.142.129	OCSP	942	Response
111	2.762616666	2.19.126.223	192.168.142.129	OCSP	942	Response
832	5.327324475	192.168.142.129	93.184.220.29	OCSP	478	[TCP Previous segment not captured] Request
846	5.506890270	93.184.220.29	192.168.142.129	OCSP	793	Response
1451	16.494818349	192.168.142.129	34.107.221.82	HTTP	399	GET / HTTP/1.1
1455	16.697354541	34.107.221.82	192.168.142.129	HTTP	284	HTTP/1.1 200 OK

Figure 5: HTTP protocol with Firefox

Hence, Mozilla Firefox also uses HTTP Protocol, and the same can be viewed in Wireshark.

Also, I have tried the same on Windows, with Chrome as browser. The obtained records are as follows

http						
No.	Time	Source	Destination	Protocol	Length	Info
15	0.335960	10.2.0.2	93.184.220.29	HTTP	276	GET /MFEwTzBNMEswSTAJBgUrDgMCGgUABBTBL0
21	0.524231	93.184.220.29	10.2.0.2	OCSP	456	Response

Figure 6: Using Windows