



Date: 01/10/2023

Lab Practical #08:

Study Packet capture and header analysis by Wireshark (TCP, UDP, IP).

Practical Assignment #08:

1. Explain usage of Wireshark tool.

Packet Capture

Wireshark captures packets in real-time as they are transmitted over a network. It can capture data from various network interfaces such as Ethernet, Wi-Fi, and others. This ability to capture packets is fundamental for analyzing network issues.

Protocol Analysis

Wireshark provides detailed information about different network protocols. It decodes the data in captured packets and presents it in a human-readable format. It supports hundreds of protocols, including common ones like TCP, UDP, HTTP, DNS, and more. This is invaluable for diagnosing network problems and understanding how applications communicate over the network.

Troubleshooting Network Issues

Network administrators use Wireshark to troubleshoot various network problems such as slow performance, connectivity issues, and suspicious activities. By analyzing the captured packets, they can identify errors, latency, or misconfigurations that might be causing problems.

Security Analysis

Wireshark is a powerful tool for security professionals. It can be used to detect security breaches, unauthorized access, and malicious activities. Security analysts use Wireshark to inspect network traffic for signs of attacks, such as malware infections, intrusion attempts, or data exfiltration.

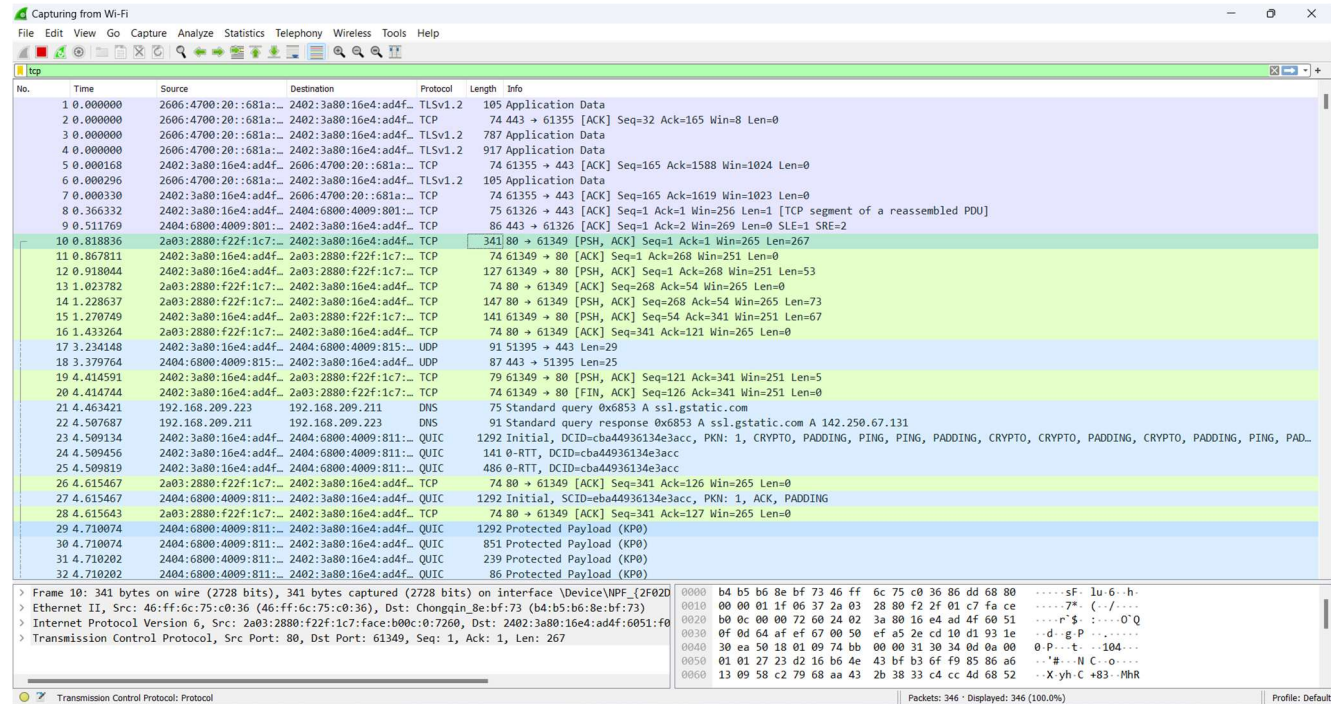
Network Performance Analysis

Wireshark helps in analyzing network performance by providing insights into data transfer rates, response times, and bottlenecks. By studying the captured packets, network administrators can optimize the network infrastructure for better performance.

Date: 01/10/2023

2. Packet capture and header analysis by Wireshark (TCP, UDP, IP).

- TCP:



Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	2606:4700:20::681a...	2402:3a80:16e4:ad4f...	TLSv1.2	105	Application Data
2	0.000000	2606:4700:20::681a...	2402:3a80:16e4:ad4f...	TCP	74	443 → 61355 [ACK] Seq=32 Ack=165 Win=8 Len=0
3	0.000000	2606:4700:20::681a...	2402:3a80:16e4:ad4f...	TLSv1.2	787	Application Data
4	0.000000	2606:4700:20::681a...	2402:3a80:16e4:ad4f...	TLSv1.2	917	Application Data
5	0.000168	2402:3a80:16e4:ad4f...	2606:4700:20::681a...	TCP	74	61355 → 443 [ACK] Seq=165 Ack=1588 Win=1024 Len=0
6	0.000296	2606:4700:20::681a...	2402:3a80:16e4:ad4f...	TLSv1.2	105	Application Data
7	0.000330	2402:3a80:16e4:ad4f...	2606:4700:20::681a...	TCP	74	61355 → 443 [ACK] Seq=165 Ack=1619 Win=1023 Len=0
8	0.366332	2402:3a80:16e4:ad4f...	2404:6800:4009:801...	TCP	75	61326 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=1 [TCP segment of a reassembled PDU]
9	0.511769	2404:6800:4009:801...	2402:3a80:16e4:ad4f...	TCP	86	443 → 61326 [ACK] Seq=1 Ack=2 Win=269 Len=0 SLE=1 SRE=2
10	0.818836	2a03:2880:f22f:1c7::...	2402:3a80:16e4:ad4f...	TCP	341	80 → 61349 [PSH, ACK] Seq=1 Ack=1 Win=265 Len=267
11	0.867811	2402:3a80:16e4:ad4f...	2a03:2880:f22f:1c7::...	TCP	74	61349 → 80 [ACK] Seq=1 Ack=268 Win=251 Len=0
12	0.918044	2402:3a80:16e4:ad4f...	2a03:2880:f22f:1c7::...	TCP	127	61349 → 80 [PSH, ACK] Seq=1 Ack=268 Win=251 Len=53
13	1.023782	2a03:2880:f22f:1c7::...	2402:3a80:16e4:ad4f...	TCP	74	80 → 61349 [ACK] Seq=268 Ack=54 Win=265 Len=0
14	1.228637	2a03:2880:f22f:1c7::...	2402:3a80:16e4:ad4f...	TCP	147	80 → 61349 [PSH, ACK] Seq=268 Ack=54 Win=265 Len=73
15	1.270749	2402:3a80:16e4:ad4f...	2a03:2880:f22f:1c7::...	TCP	141	61349 → 80 [PSH, ACK] Seq=54 Ack=341 Win=251 Len=67
16	1.433264	2a03:2880:f22f:1c7::...	2402:3a80:16e4:ad4f...	TCP	74	80 → 61349 [ACK] Seq=341 Ack=121 Win=265 Len=0
17	3.234148	2402:3a80:16e4:ad4f...	2404:6800:4009:815...	UDP	91	51395 → 443 Len=29
18	3.379764	2404:6800:4009:815...	2402:3a80:16e4:ad4f...	UDP	87	443 → 51395 Len=25
19	4.414591	2402:3a80:16e4:ad4f...	2a03:2880:f22f:1c7::...	TCP	79	61349 → 80 [PSH, ACK] Seq=121 Ack=341 Win=251 Len=5
20	4.414744	2402:3a80:16e4:ad4f...	2a03:2880:f22f:1c7::...	TCP	74	61349 → 80 [FIN, ACK] Seq=126 Ack=341 Win=251 Len=0
21	4.463421	192.168.209.223	192.168.209.211	DNS	75	Standard query 0x6853 A ssl.gstatic.com
22	4.507687	192.168.209.211	192.168.209.223	DNS	91	Standard query response 0x6853 A ssl.gstatic.com A 142.250.67.131
23	4.509134	2402:3a80:16e4:ad4f...	2404:6800:4009:811...	QUIC	1292	Initial, DCID=eba44936134e3acc, PKN: 1, CRYPTO, PADDING, PING, PING, PADDING, CRYPTO, CRYPTO, PADDING, CRYPTO, PADDING, PING, PAD...
24	4.509456	2402:3a80:16e4:ad4f...	2404:6800:4009:811...	QUIC	141	0-RTT, DCID=eba44936134e3acc
25	4.509819	2402:3a80:16e4:ad4f...	2404:6800:4009:811...	QUIC	486	0-RTT, DCID=eba44936134e3acc
26	4.615467	2a03:2880:f22f:1c7::...	2402:3a80:16e4:ad4f...	TCP	74	80 → 61349 [ACK] Seq=341 Ack=126 Win=265 Len=0
27	4.615467	2404:6800:4009:811...	2402:3a80:16e4:ad4f...	QUIC	1292	Initial, SCID=eba44936134e3acc, PKN: 1, ACK, PADDING
28	4.615643	2a03:2880:f22f:1c7::...	2402:3a80:16e4:ad4f...	TCP	74	80 → 61349 [ACK] Seq=341 Ack=127 Win=265 Len=0
29	4.710074	2404:6800:4009:811...	2402:3a80:16e4:ad4f...	QUIC	1292	Protected Payload (KP0)
30	4.710074	2404:6800:4009:811...	2402:3a80:16e4:ad4f...	QUIC	851	Protected Payload (KP0)
31	4.710202	2404:6800:4009:811...	2402:3a80:16e4:ad4f...	QUIC	239	Protected Payload (KP0)
32	4.710202	2404:6800:4009:811...	2402:3a80:16e4:ad4f...	QUIC	86	Protected Payload (KP0)

Packet Details:

Frame 10: 341 bytes on wire (2728 bits), 341 bytes captured (2728 bits) on interface \Device\NPF_{2F02...}

Ethernet II, Src: 46:ff:6c:75:c0:36 (46:ff:6c:75:c0:36), Dst: Chongqin_8e:bf:73 (b4:b5:b6:8e:bf:73)

Internet Protocol Version 6, Src: 2a03:2880:f22f:1c7::face:b00c:0:7260, Dst: 2402:3a80:16e4:ad4f:6051:f0...

Transmission Control Protocol, Src Port: 80, Dst Port: 61349, Seq: 1, Ack: 1, Len: 267

Packet Bytes:

```

0000  b4 b5 b6 8e bf 73 46 ff  6c 75 c0 36 8d dd 68 80  ....sF..lu.6..h.
0010  00 00 01 1f 06 37 2a 03  28 80 f2 2f 01 c7 fa ce  ....7*..(-/....
0020  b0 0c 00 00 72 60 24 02  3a 80 16 e4 ad 4f 60 51  ....r$. ....O'Q
0030  0f 0d 64 af ef 67 00 50  ef a5 2e cd 10 d1 93 1e  ..d.gP.....
0040  30 ea 50 18 01 09 74 bb  00 00 31 30 34 0d 0a 00  0.P...t...104...
0050  01 01 27 23 d2 16 b6 4e  43 bf b3 6f f9 85 86 a6  ...#.N C.o....
0060  13 09 58 c2 79 68 aa 43  2b 38 33 c4 cc 4d 68 52  ..X.y.C +83..JhR
  
```



DARSHAN INSTITUTE OF ENGINEERING & TECHNOLOGY

Semester 5th | Practical Assignment | Computer Networks (2101CS501)

Date: 01/10/2023

• UDP:

Wireshark packet capture showing UDP traffic. The packet list shows a series of UDP packets from 2402:3a80:16e4:ad4f to 2404:6800:4009:8115. The packet details pane shows the structure of a UDP packet, including the header and payload. The packet bytes pane shows the raw data in hexadecimal and ASCII.

• IP:

Wireshark packet capture showing IP traffic. The packet list shows a series of IP packets from 2402:3a80:16e4:ad4f to 2404:6800:4009:8115. The packet details pane shows the structure of an IP packet, including the header and payload. The packet bytes pane shows the raw data in hexadecimal and ASCII.