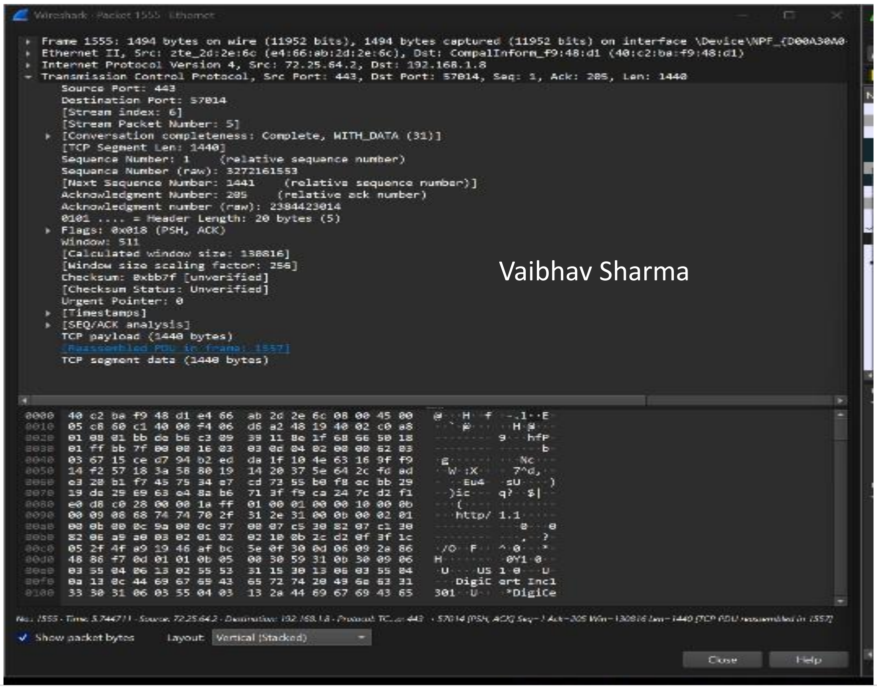
Experiment No: 12	
Name	Vaibhav Sharma
PRN	22070126125
Date of Performance	16th October 2024
Title	Implement and analyze TCP and ICMP protocols using Wireshark
Objective	The purpose of this experiment is to utilize Wireshark to capture and analyse **TCP** and **ICMP** packets. The experiment aims to understand the structure and behaviour of these protocols during network communication.
Setup	• Wireshark was installed and configured to monitor the Ethernet interface. • The focus was on capturing **TCP** and **ICMP** traffic, applying respective filters in Wireshark.
Procedure	**Step 1: Capturing TCP Packets** 1. A packet capture session was initiated on the Ethernet interface. 2. **TCP** filters were applied to narrow the capture to TCP traffic. 3. A TCP connection to a web server (port 443) was established, capturing the packets exchanged during the session. **Step 2: Capturing ICMP Packets** 1. A new capture session was started, and the **ICMP** filter was applied. 2. **Ping** commands were issued to a remote server (www.google.com) to generate ICMP Echo Requests and Responses.

Analysis	TCP Packet Analysis • **Frame 1555** represents a **TCP segment** associated with encrypted application data over **TLSv1.2** • Source IP: 72.25.64.2 • Destination IP: 192.168.1.8 • Source Port: 443 (HTTPS) • Destination Port: 57014 (Client) • TCP Flags: PSH, ACK • Window Size: 130816 • Sequence Number: 3727216553 • Acknowledgment Number: 2348424014 • Payload Length: 1440 bytes ICMP Packet Analysis • **Frame 52** represents an **ICMP Echo Request** packet sent during a ping • Source IP: 2404:6800:4009:824::2004 (Google Server) • Destination IP: 2409:40e2:102d:c3d3:49e2:37cf:2c0b (Local Machine) • Protocol: ICMPv6 • Ping Request ID: 0x0001 • Sequence Number: 7 • ICMP Data: 32 bytes • The response (Frame 53) was received, confirming successful communication with a round-trip time of 39ms.

Screenshots

1. TCP



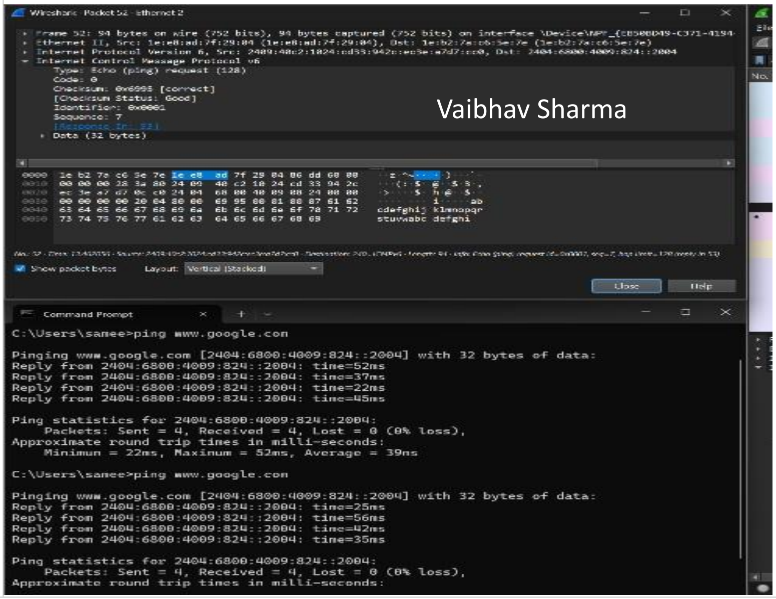
```
Wireshark · Packet 1555 · Ethernet

▸ Frame 1555: 1494 bytes on wire (11952 bits), 1494 bytes captured (11952 bits) on interface \Device\NPF_{D00A30A0-
▸ Ethernet II, Src: zte_2d:2e:6c (e4:66:ab:2d:2e:6c), Dst: CompalInform_f9:48:d1 (40:c2:ba:f9:48:d1)
▸ Internet Protocol Version 4, Src: 72.25.64.2, Dst: 192.168.1.8
▾ Transmission Control Protocol, Src Port: 443, Dst Port: 57014, Seq: 1, Ack: 205, Len: 1440
     Source Port: 443
     Destination Port: 57014
     [Stream index: 6]
     [Stream Packet Number: 5]
   ▸ [Conversation completeness: Complete, WITH_DATA (31)]
     [TCP Segment Len: 1440]
     Sequence Number: 1    (relative sequence number)
     Sequence Number (raw): 3272161553
     [Next Sequence Number: 1441    (relative sequence number)]
     Acknowledgment Number: 205    (relative ack number)
     Acknowledgment number (raw): 2384423014
     0101 .... = Header Length: 20 bytes (5)
   ▸ Flags: 0x018 (PSH, ACK)
     Window: 511
     [Calculated window size: 130816]
     [Window size scaling factor: 256]
     Checksum: 0xbb7f [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   ▸ [Timestamps]
   ▸ [SEQ/ACK analysis]
     TCP payload (1440 bytes)
     [Reassembled PDU in frame: 1557]
     TCP segment data (1440 bytes)

0000  40 c2 ba f9 48 d1 e4 66  ab 2d 2e 6c 08 00 45 00   @···H··f ·-.l··E·
0010  05 c8 60 c1 40 00 f4 06  d6 a2 48 19 40 02 c0 a8   ··`·@··· ··H·@···
0020  01 08 01 bb de b6 c3 09  39 11 8e 1f 68 66 50 18   ········ 9···hfP·
0030  01 ff bb 7f 00 00 16 03  03 0d 04 02 00 00 62 03   ········ ······b·
0040  03 67 15 ce d7 94 b2 ed  da 1f 10 4e 63 16 9f f9   ·g······ ···Nc···
0050  14 f2 57 18 3a 58 80 19  14 20 37 5e 64 2c fd ad   ··W·:X·· · 7^d,··
0060  e3 20 b1 f7 45 75 34 e7  cd 73 55 b0 f8 ec bb 29   · ··Eu4· ·sU····)
0070  19 de 29 69 63 e4 8a b6  71 3f f9 ca 24 7c d2 f1   ··)ic··· q?··$|··
0080  e0 d8 c0 28 00 00 1a ff  01 00 01 00 00 10 00 0b   ···(···· ········
0090  00 09 08 68 74 74 70 2f  31 2e 31 00 0b 00 02 01   ···http/ 1.1·····
00a0  00 0b 00 0c 9a 00 0c 97  00 07 c5 30 82 07 c1 30   ········ ···0···0
00b0  82 06 a9 a0 03 02 01 02  02 10 0b 2c d2 0f 3f 1c   ········ ···,··?·
00c0  05 2f 4f a9 19 46 af bc  5e 0f 30 0d 06 09 2a 86   ·/O··F·· ^·0···*·
00d0  48 86 f7 0d 01 01 0b 05  00 30 59 31 0b 30 09 06   H······· ·0Y1·0··
00e0  03 55 04 06 13 02 55 53  31 15 30 13 06 03 55 04   ·U····US 1·0···U·
00f0  0a 13 0c 44 69 67 69 43  65 72 74 20 49 6e 63 31   ···DigiC ert Inc1
0100  33 30 31 06 03 55 04 03  13 2a 44 69 67 69 43 65   301··U·· ·*DigiCe

No.: 1555 · Time: 5.744711 · Source: 72.25.64.2 · Destination: 192.168.1.8 · Protocol: TC...or 443 → 57014 [PSH, ACK] Seq=1 Ack=205 Win=130816 Len=1440 [TCP PDU reassembled in 1557]
```

2. ICMP



```
Wireshark · Packet 52 · Ethernet 2

▸ Frame 52: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface \Device\NPF_{CB500D49-C371-4194-
▸ Ethernet II, Src: 1e:e8:ad:7f:29:04 (1e:e8:ad:7f:29:04), Dst: 1e:b2:7a:c6:5e:7e (1e:b2:7a:c6:5e:7e)
▸ Internet Protocol Version 6, Src: 2409:40c2:1024:cd33:942c:ec3e:a7d7:cc0, Dst: 2404:6800:4009:824::2004
▾ Internet Control Message Protocol v6
     Type: Echo (ping) request (128)
     Code: 0
     Checksum: 0x6995 [correct]
     [Checksum Status: Good]
     Identifier: 0x0001
     Sequence: 7
     [Response In: 53]
   ▸ Data (32 bytes)

0000  1e b2 7a c6 5e 7e 1e e8  ad 7f 29 04 86 dd 60 00   ··z·^~·· ··)···`·
0010  00 00 00 28 3a 80 24 09  40 c2 10 24 cd 33 94 2c   ···(:·$· @··$·3·,
0020  ec 3e a7 d7 0c c0 24 04  68 00 40 09 08 24 00 00   ·>····$· h·@··$··
0030  00 00 00 00 20 04 80 00  69 95 00 01 00 07 61 62   ···· ··· i·····ab
0040  63 64 65 66 67 68 69 6a  6b 6c 6d 6e 6f 70 71 72   cdefghij klmnopqr
0050  73 74 75 76 77 61 62 63  64 65 66 67 68 69         stuvwabc defghi

No.: 52 · Time: 13.462036 · Source: 2409:40c2:1024:cd33:942c:ec3e:a7d7:cc0 · Destination: 240...:2004 · Length: 94 · Info: Echo (ping) request id=0x0001, seq=7, hop limit=128 (reply in 53)
```

```
C:\Users\samee>ping www.google.com

Pinging www.google.com [2404:6800:4009:824::2004] with 32 bytes of data:
Reply from 2404:6800:4009:824::2004: time=52ms
Reply from 2404:6800:4009:824::2004: time=37ms
Reply from 2404:6800:4009:824::2004: time=22ms
Reply from 2404:6800:4009:824::2004: time=45ms

Ping statistics for 2404:6800:4009:824::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 22ms, Maximum = 52ms, Average = 39ms

C:\Users\samee>ping www.google.com

Pinging www.google.com [2404:6800:4009:824::2004] with 32 bytes of data:
Reply from 2404:6800:4009:824::2004: time=25ms
Reply from 2404:6800:4009:824::2004: time=56ms
Reply from 2404:6800:4009:824::2004: time=42ms
Reply from 2404:6800:4009:824::2004: time=35ms

Ping statistics for 2404:6800:4009:824::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

Observation	The TCP packets captured were part of an encrypted session between a client and server over HTTPS. The PSH, ACK flags were used to ensure data was sent and acknowledged efficiently. The ICMP packets were generated from ping requests, and responses confirmed connectivity and the average round-trip time (RTT) to the server was 39ms.
Self-assessment Q&A	Q: What is the purpose of analyzing TCP and ICMP protocols using Wireshark? Ans: To understand how TCP manages reliable data transmission and how ICMP handles error reporting and diagnostic messages during network communication. Q: How does Wireshark capture TCP packets? Ans: Wireshark captures TCP packets by monitoring communication on specific ports, such as 80 for HTTP or 443 for HTTPS, and displays sequence numbers, acknowledgments, and flags. Q: What information can be gathered from ICMP packet analysis? Ans: ICMP packet analysis provides details on network diagnostics, such as ping requests and replies, as well as error messages like destination unreachable or time exceeded.
Conclusion	This experiment allowed the exploration of both **TCP** and **ICMP** protocols using Wireshark. It provided insight into how TCP handles reliable transmission and how ICMP helps in diagnosing network connectivity through ping. The ability to capture and analyse the packet structure, including flags, sequence numbers, and headers, furthered the understanding of these crucial network protocols.