

## SYS-207 – Networks (Assessment Only)

**00070 Assessment - Understand how transport protocols do protocol multiplexing, division into segments and reassembly of long messages, end-to-end performance control, assurance of (a) at-least-once delivery; (b) at-most-once delivery; (c) data**

Name: Thanawin Pattanaphol

ID: 01324096

## Objective

Capture a traffic trace with Wireshark and identify where three End-To-End-Layer techniques appear in the trace, then, interpret one of those techniques in detail using the official protocol standards.

## Captured Trace

The trace was captured on the eno1 interface, in this context, the Wi-Fi card; shown in Figure 1.

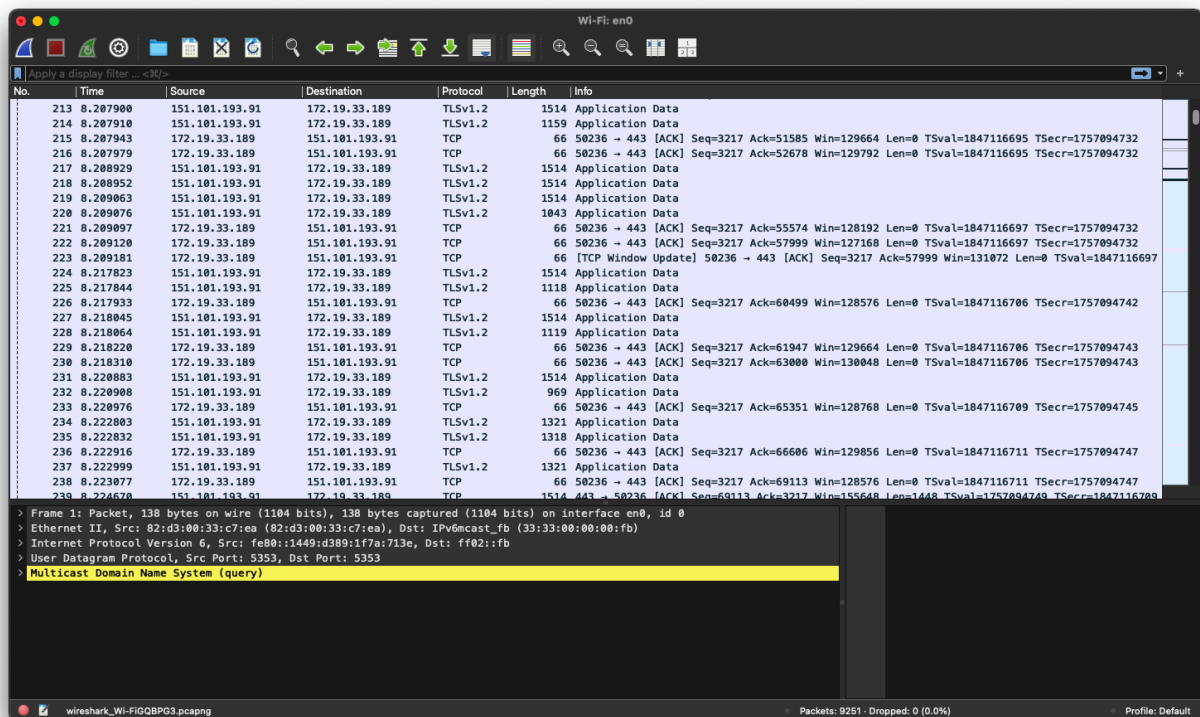


Figure 1

## End-To-End-Layer Techniques

To demonstrate the different End-To-End-Layer techniques shown in the Wireshark trace, this report will display three different techniques that can be found in this trace.

### Connection establishment (TCP three-way handshake)

The Figure 2 shows a filtered Wireshark trace with the filter of `tcp.flags.syn == 1 || tcp.flags.fin == 1`, this filters packets such as [SYN], [SYN, ACK], and [ACK]

No.	Time	Source	Destination	Protocol	Length	Info
36	7.662670	104.21.20.105	172.19.33.189	TCP	66	443 → 50229 [FIN, ACK] Seq=80 Ack=92 Win=
39	7.667739	172.19.33.189	151.101.65.91	TCP	78	50234 → 443 [SYN, ECE, CWR] Seq=0 Win=655
41	7.667958	172.19.33.189	104.21.20.105	TCP	66	50229 → 443 [FIN, ACK] Seq=116 Ack=81 Win=
43	7.671480	151.101.65.91	172.19.33.189	TCP	74	443 → 50234 [SYN, ACK] Seq=0 Ack=1 Win=65
45	7.677701	172.19.33.189	151.101.65.91	TCP	78	50235 → 443 [SYN, ECE, CWR] Seq=0 Win=655

```

> Frame 43: Packet, 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface en0, id 0
> Ethernet II, Src: 62:a7:79:9a:7b:d4 (62:a7:79:9a:7b:d4), Dst: Apple_d8:d0:cb (98:01:a7:d8:d0:cb)
> Internet Protocol Version 4, Src: 151.101.65.91, Dst: 172.19.33.189
> Transmission Control Protocol, Src Port: 443, Dst Port: 50234, Seq: 0, Ack: 1, Len: 0
  Source Port: 443
  Destination Port: 50234
  [Stream index: 1]
  [Stream Packet Number: 2]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0 (relative sequence number)
  Sequence Number (raw): 2791431454
  [Next Sequence Number: 1 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 239799893
  1010 .... = Header Length: 40 bytes (10)
  > Flags: 0x012 (SYN, ACK)
    Window: 65535
    [Calculated window size: 65535]
    Checksum: 0xacac [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
  > [Timestamps]
  > [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]

```

Figure 2

The three-way handshake synchronizes sequences numbers (ISS/IRS) and brings the endpoints to the ESTABLISHED TCP state; it prevents old duplicate segments from causing false connections [1].

## Segmentation & Reassembly of long messages

Time	Source	Destination	Protocol	Length	Info
73	7.689966	151.101.65.91	172.19.33.189	TCP	66 443 → 50234 [ACK] Seq=1550 Ack=3601 Win=152576
74	7.690431	172.19.33.189	151.101.193.91	TCP	1514 50236 → 443 [ACK] Seq=1 Ack=1 Win=131776 Len=14
75	7.690433	172.19.33.189	151.101.193.91	TLSv1.2	970 Client Hello (SNI=api.accounts.firefox.com)

```
> Frame 74: Packet, 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: Apple_d8:d0:cb (98:01:a7:d8:d0:cb), Dst: 62:a7:79:9a:7b:d4 (62:a7:79:9a:7b:d4)
> Internet Protocol Version 4, Src: 172.19.33.189, Dst: 151.101.193.91
> Transmission Control Protocol, Src Port: 50236, Dst Port: 443, Seq: 1, Ack: 1, Len: 1448
  Source Port: 50236
  Destination Port: 443
  [Stream index: 3]
  [Stream Packet Number: 4]
  > [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 1448]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3696114775
  [Next Sequence Number: 1449 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2934957499
  1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
  Window: 2059
  [Calculated window size: 131776]
  [Window size scaling factor: 64]
  Checksum: 0xa5e8 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
  TCP payload (1448 bytes)
  [Reassembled PDU in frame: 75]
  TCP segment data (1448 bytes)
```

Figure 3

Figure 3 displays a reassembled packet, a demonstration of a large application payload across multiple TCP segments because of MSS/MTU limitations. The receiver uses sequences number and the TCP stream to reassemble the contiguous byte stream before delivering it to the higher-level protocol. This is an end-to-end functionality implemented by TCP and used by application protocols which is described in RFC 793 standard [2].

## Retransmission (loss recovery to at-least-once delivery behavior)

Figure 4 shows a TCP Retransmission packet, this happens when TCP detects missing ACKs and retransmits the unacknowledged segment to ensure delivery. This provides the reliability property. RFC 973 discusses retransmission expectations and sequence number handling. [3]

No.	Time	Source	Destination	Protocol	Length	Info
84	7.696170	151.101.193.91	172.19.33.189	TCP	1514	[TCP Retransmission] 443 → 50236 [ACK] Seq=2934957499
196	8.193716	151.101.193.91	172.19.33.189	TCP	364	[TCP Retransmission] 443 → 50238 [PSH, ACK] Seq=2934957499
286	8.876308	151.101.193.91	172.19.33.189	TCP	1514	[TCP Retransmission] 443 → 50238 [ACK] Seq=2934957499
298	9.358861	172.19.33.189	104.21.20.105	TCP	97	[TCP Retransmission] 50239 → 443 [PSH, ACK] Seq=3696117127
300	9.524475	104.21.20.105	172.19.33.189	TCP	93	[TCP Retransmission] 443 → 50239 [PSH, ACK] Seq=3696117127
1050	12.386287	151.101.193.91	172.19.33.189	TCP	1514	[TCP Retransmission] 443 → 50238 [ACK] Seq=2934957499

> Frame 84: Packet, 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface en0, id 0
> Ethernet II, Src: 62:a7:79:9a:7b:d4 (62:a7:79:9a:7b:d4), Dst: Apple_d8:d0:cb (98:01:a7:d8:d0:cb)
> Internet Protocol Version 4, Src: 151.101.193.91, Dst: 172.19.33.189
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 50236, Seq: 1, Ack: 2353, Len: 1448
Source Port: 443
Destination Port: 50236
[Stream index: 3]
[Stream Packet Number: 8]
> [Conversation completeness: Incomplete, DATA (15)]
[TCP Segment Len: 1448]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2934957499
[Next Sequence Number: 1449 (relative sequence number)]
Acknowledgment Number: 2353 (relative ack number)
Acknowledgment number (raw): 3696117127
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x010 (ACK)
Window: 292
[Calculated window size: 149504]
[Window size scaling factor: 512]
Checksum: 0xebec [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
> [Timestamps]
> [SEQ/ACK analysis]
[Client Contiguous Streams: 1]
[Server Contiguous Streams: 1]
TCP payload (1448 bytes)
Retransmitted TCP segment data (1448 bytes)

Figure 4

## References

- [1] J. Postel, *Transmission Control Protocol*, RFC 793, Internet Engineering Task Force (IETF), Sep. 1981. (See §3.4 “Establishing a Connection”, pp. 27–31 for TCP three-way handshake and ISN synchronization.)
- [2] J. Postel, *User Datagram Protocol*, RFC 768, Internet Engineering Task Force (IETF), Aug. 1980. (UDP transport multiplexing description.)
- [3] J. Postel, *Internet Protocol*, RFC 791, IETF, Sep. 1981. (End-to-end fragmentation and reassembly rules — pp. 13–23.)