# CSC3511: TCP Wireshark

**Name:** Austin Koske

---

## 1. HTTP Packet Analysis

1. Fire up Wireshark and start capturing. Browse to a webpage (e.g. `http://httpvshttps.com`).
   Filter with the `http` filter and select the first packet in the request.
   Turn off *auto-scroll*.
   Now clear the `http` filter and press ENTER — the same packet should remain selected.
   If helpful, right-click on the packet and select **Follow → TCP Stream** to isolate packets for that connection.
   You may also use a filter like:

   ```
   ip.addr == 45.33.7.16
   ```

   using your server's IP address.

---

### a. TCP Ports

Identify where the TCP source and destination ports appear within the hexadecimal shorthand packet data.
Look at a TCP connection to a web server.

**Write the destination port (on the server) in:** - Decimal: `80` - Hexadecimal: `0x50`

---

### b. (If you have time) TCP Sequence and Acknowledgement Numbers

Identify the **TCP sequence number** and **acknowledgement number** in your packet.

**Write these numbers (in hexadecimal only):**
- Sequence number: `0` - Acknowledgement number: `0`

---

### c. (If you have time) Maximum TCP Source Port

Determine the **maximum value** of the TCP source port.

**Answer:** `65535`

**Source**

### d. (If you have time) Maximum TCP Sequence Number

Determine, approximately, the **maximum value** of the TCP sequence number.

**Answer:** `4,294,967,295` = $(2^{32} - 1)$

## 2. SYN and ACK Messages

### a.

Identify the **SYN packet** sent from the client to the server.
**Sequence number:** `0`

### b. (If you have time)

Identify the **SYN packet** sent from the server to the client in response.
Does the packet have the **SYN value** you expect?
**Answer:** Yes. The server replied with **SYN + ACK** (`0x012`)

### c. (If you have time)

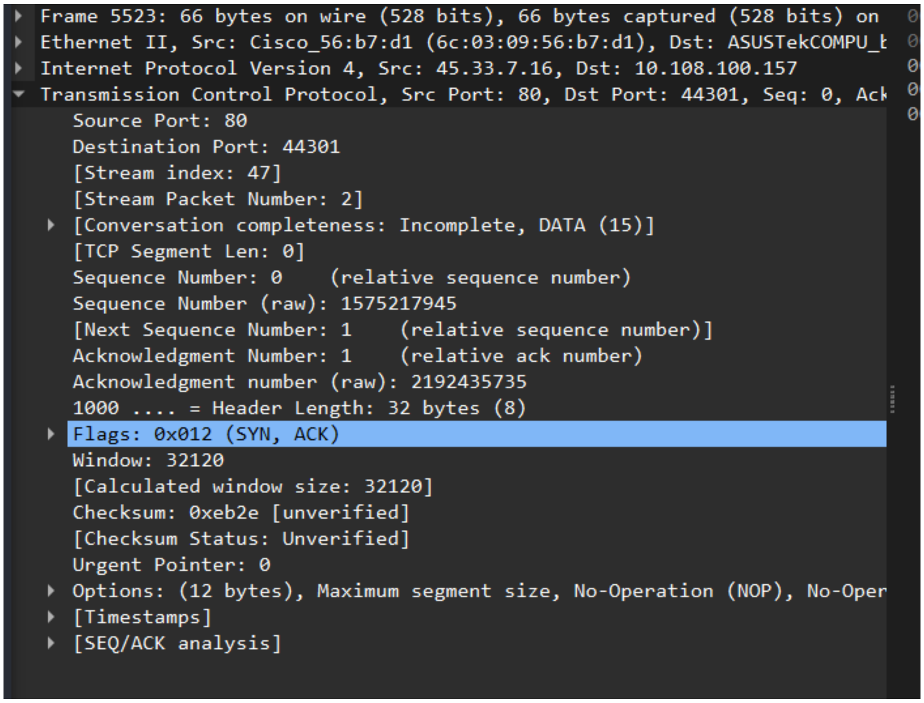
Identify the **second packet** from the client to the server. Does it have the **SYN and ACK** values you expect? **Answer:** Yes. The second packet from the **client** has only the **ACK** flag set (`0x010`), which is expected. This confirms the final step of the TCP three-way handshake — the client acknowledges the server's **SYN + ACK**.

### d. (If you have time)

Identify the **SYN and ACK fields** within the TCP header.
Repeat the above exercises considering the **actual values** rather than Wireshark's interpreted ones.
**Answer:**

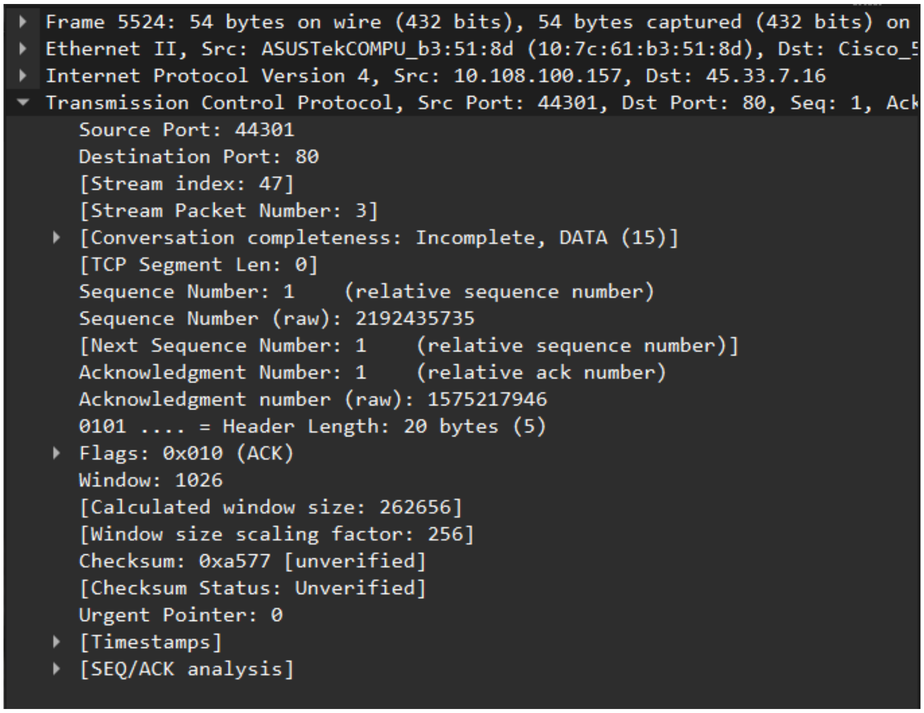
### e. (If you have time)

Can you see any other **TCP packets** to the same server?
**Answer:** Yes, there is one HTTP GET request, an ACK to that request, one HTTP GET response, and one ACK to that response.



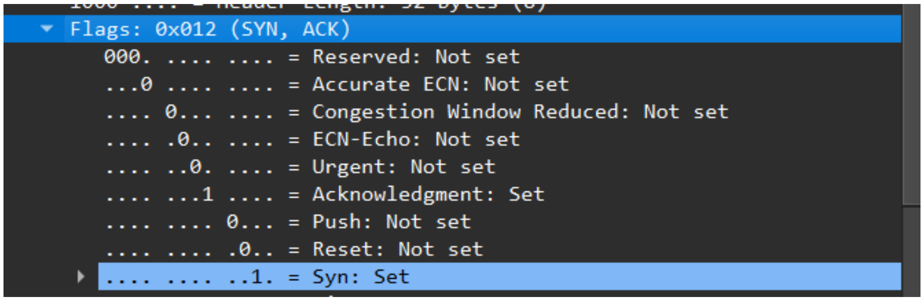
```
▸ Frame 5523: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on
▸ Ethernet II, Src: Cisco_56:b7:d1 (6c:03:09:56:b7:d1), Dst: ASUSTekCOMPU_t
▸ Internet Protocol Version 4, Src: 45.33.7.16, Dst: 10.108.100.157
▾ Transmission Control Protocol, Src Port: 80, Dst Port: 44301, Seq: 0, Ack
     Source Port: 80
     Destination Port: 44301
     [Stream index: 47]
     [Stream Packet Number: 2]
   ▸ [Conversation completeness: Incomplete, DATA (15)]
     [TCP Segment Len: 0]
     Sequence Number: 0    (relative sequence number)
     Sequence Number (raw): 1575217945
     [Next Sequence Number: 1    (relative sequence number)]
     Acknowledgment Number: 1    (relative ack number)
     Acknowledgment number (raw): 2192435735
     1000 .... = Header Length: 32 bytes (8)
   ▸ Flags: 0x012 (SYN, ACK)
     Window: 32120
     [Calculated window size: 32120]
     Checksum: 0xeb2e [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   ▸ Options: (12 bytes), Maximum segment size, No-Operation (NOP), No-Oper
   ▸ [Timestamps]
   ▸ [SEQ/ACK analysis]
```

Figure 1: image



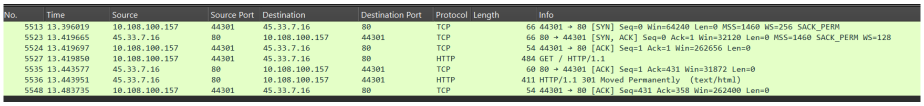
```
▸ Frame 5524: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on
▸ Ethernet II, Src: ASUSTekCOMPU_b3:51:8d (10:7c:61:b3:51:8d), Dst: Cisco_5
▸ Internet Protocol Version 4, Src: 10.108.100.157, Dst: 45.33.7.16
▾ Transmission Control Protocol, Src Port: 44301, Dst Port: 80, Seq: 1, Ack
     Source Port: 44301
     Destination Port: 80
     [Stream index: 47]
     [Stream Packet Number: 3]
   ▸ [Conversation completeness: Incomplete, DATA (15)]
     [TCP Segment Len: 0]
     Sequence Number: 1    (relative sequence number)
     Sequence Number (raw): 2192435735
     [Next Sequence Number: 1    (relative sequence number)]
     Acknowledgment Number: 1    (relative ack number)
     Acknowledgment number (raw): 1575217946
     0101 .... = Header Length: 20 bytes (5)
   ▸ Flags: 0x010 (ACK)
     Window: 1026
     [Calculated window size: 262656]
     [Window size scaling factor: 256]
     Checksum: 0xa577 [unverified]
     [Checksum Status: Unverified]
     Urgent Pointer: 0
   ▸ [Timestamps]
   ▸ [SEQ/ACK analysis]
```

Figure 2: image



```
    1000 .... = Header Length: 32 bytes (8)
  ▾ Flags: 0x012 (SYN, ACK)
       000. .... .... = Reserved: Not set
       ...0 .... .... = Accurate ECN: Not set
       .... 0... .... = Congestion Window Reduced: Not set
       .... .0.. .... = ECN-Echo: Not set
       .... ..0. .... = Urgent: Not set
       .... ...1 .... = Acknowledgment: Set
       .... .... 0... = Push: Not set
       .... .... .0.. = Reset: Not set
     ▸ .... .... ..1. = Syn: Set
```

Figure 3: image



| No. | Time | Source | Source Port | Destination | Destination Port | Protocol | Length | Info |
|---|---|---|---|---|---|---|---|---|
| 5513 | 13.396019 | 10.108.100.157 | 44301 | 45.33.7.16 | 80 | TCP | 66 | 44301 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM |
| 5523 | 13.419665 | 45.33.7.16 | 80 | 10.108.100.157 | 44301 | TCP | 66 | 80 → 44301 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128 |
| 5524 | 13.419697 | 10.108.100.157 | 44301 | 45.33.7.16 | 80 | TCP | 54 | 44301 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0 |
| 5527 | 13.419850 | 10.108.100.157 | 44301 | 45.33.7.16 | 80 | HTTP | 484 | GET / HTTP/1.1 |
| 5535 | 13.443577 | 45.33.7.16 | 80 | 10.108.100.157 | 44301 | TCP | 60 | 80 → 44301 [ACK] Seq=1 Ack=431 Win=31872 Len=0 |
| 5536 | 13.443951 | 45.33.7.16 | 80 | 10.108.100.157 | 44301 | HTTP | 411 | HTTP/1.1 301 Moved Permanently  (text/html) |
| 5548 | 13.483735 | 10.108.100.157 | 44301 | 45.33.7.16 | 80 | TCP | 54 | 44301 → 80 [ACK] Seq=431 Ack=358 Win=262400 Len=0 |

Figure 4: image

**f. (If you have time)**

Explore the **other fields** in the packet.
**Questions you have about them:**
- What are the other flags? (Reserved, Accurate, Urgent, etc)

## 3. TCP with Stop-and-Wait

Fill in the blanks in the following TCP stream.
*(The numbers are above the arrows they describe.)*

## 5. TCP with Pipelined Sliding Window

Fill in the blanks below.
*(The numbers are above the arrows they describe.)*

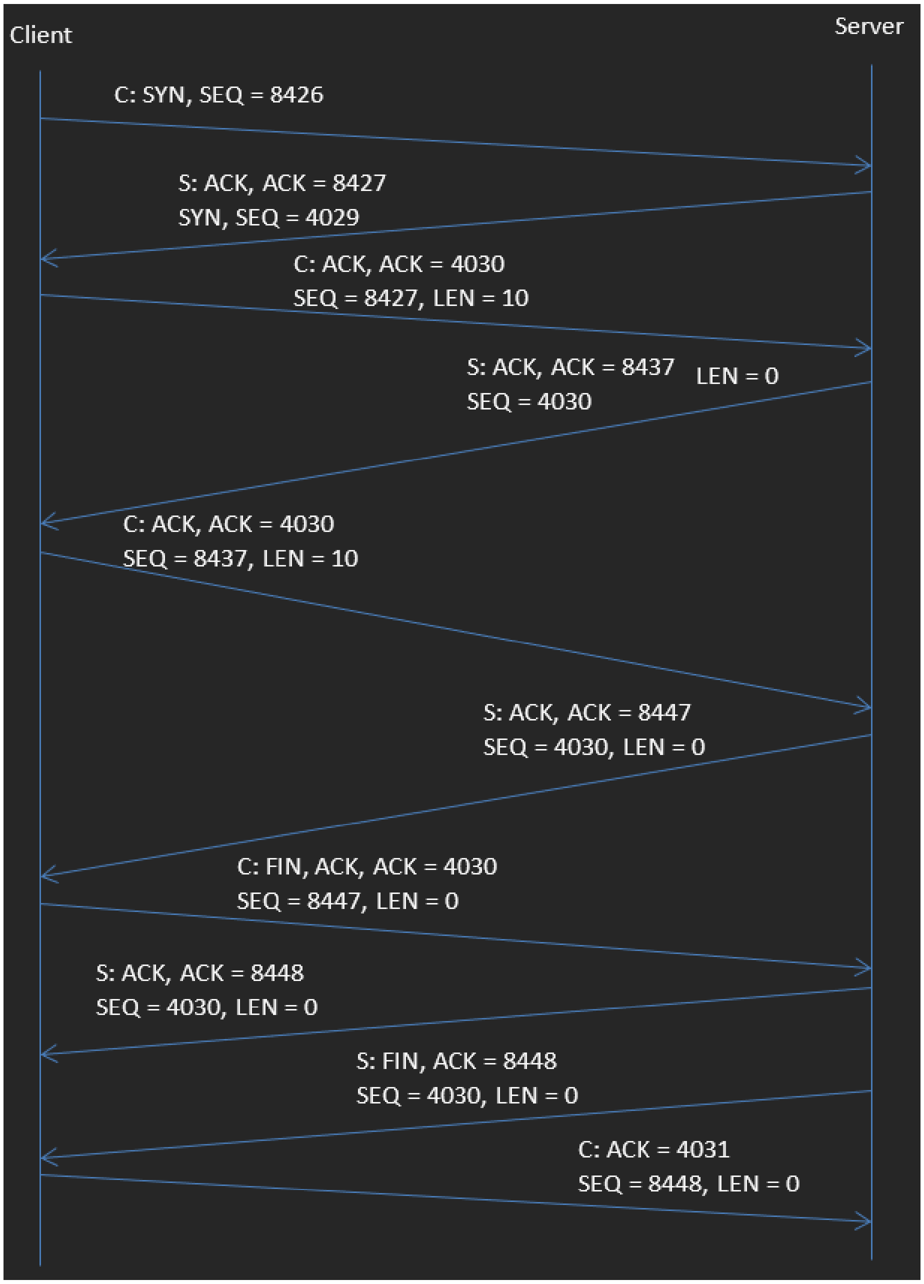


Figure 5: image

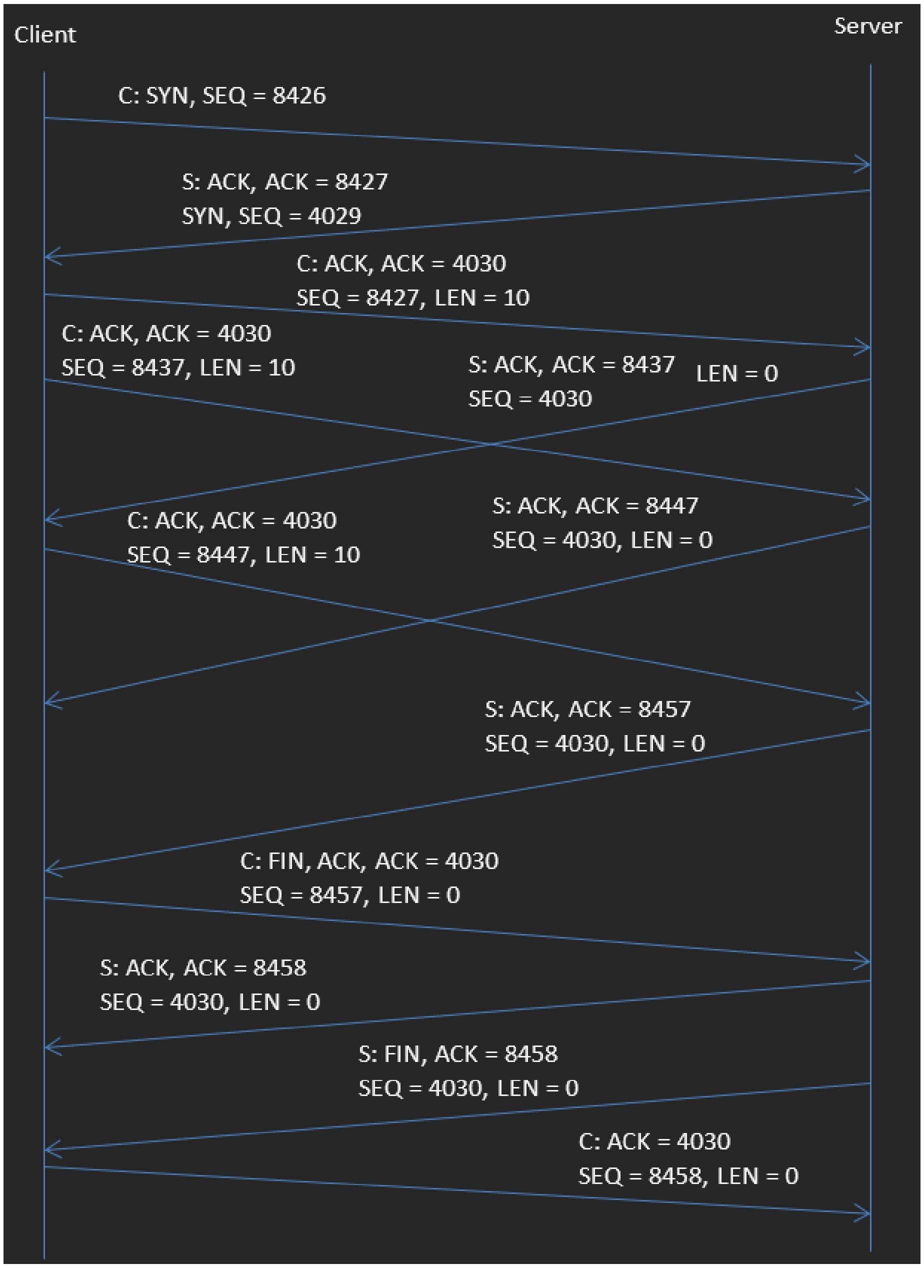


Figure 6: image