TCP Packet Capture Analysis

4 years ago
by Bamdeb Ghosh

What is TCP?

TCP (Transmission Control Protocol) is a connection oriented transport layer protocol.

Intention of this article:

To understand whole TCP is not an easy task. In this article we will try to understand the basic packets exchanges of TCP through Wireshark. Theory can be read through internet. We will focus more on packet capture analysis.

Why TCP is famous?

There are multiple reasons why TCP is so famous:

0 of 8 minutes, 22 secondsVolume 0%

1. TCP is connection orientated protocol so reliability is very high.
2. TCP can control congestion by itself.
3. TCP can detect error.
4. TCP uses flow control protocol.
5. TCP has delay ACK features.
6. TCP has selective ACK feature.
7. TCP has windows calling feature for throughput improvement.

There are so many other features that make TCP so famous.

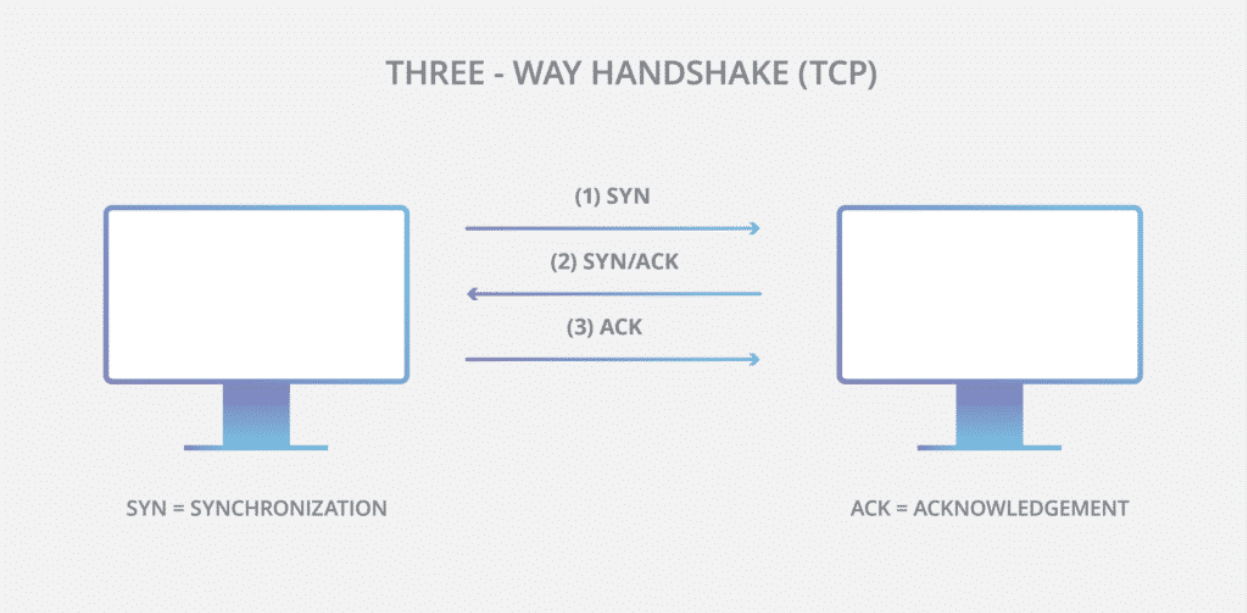
Analysis of TCP:

We will follow some steps to generate TCP frames.

Step 1: The simple way to generate TCP packets is by accessing any HTTP website. The reason is, HTTP is an application layer protocol and it uses TCP as underlying transport layer protocol.
To know about HTTP follow below link

https://linuxhint.com/http_wireshark/

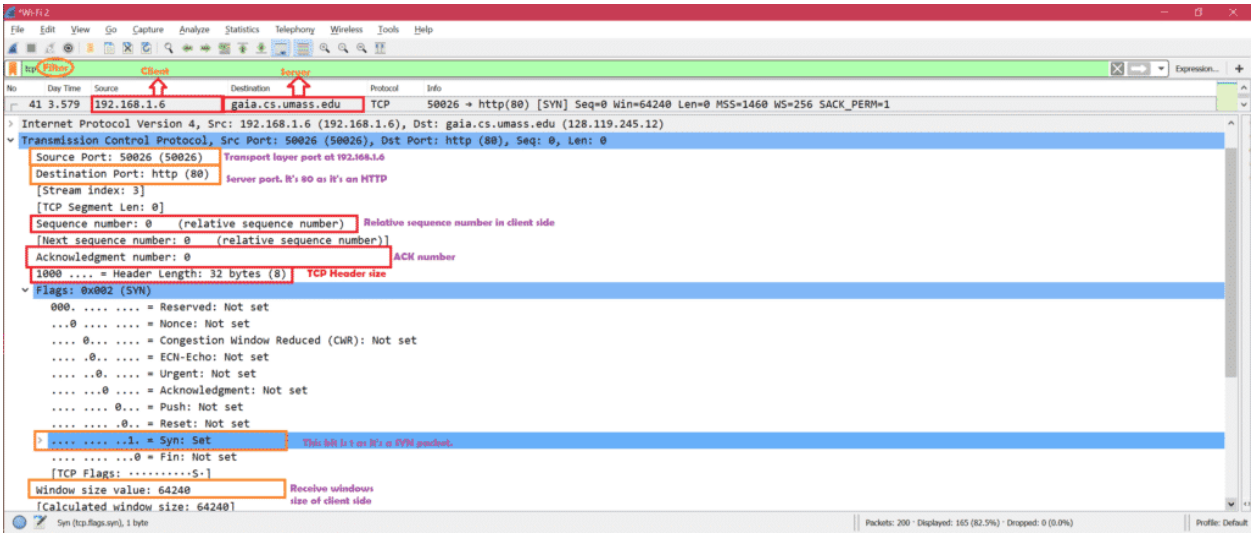
Step 2: Start Wireshark.
Step 3: Open below link in any browser.
http://gaia.cs.umass.edu/wireshark-labs/alice.txt
Step 4: Stop Wireshark and put TCP as filter.
Step 5: ANALYSIS
Now we should see TCP 3-way handshake packets. Here is the simple diagram.

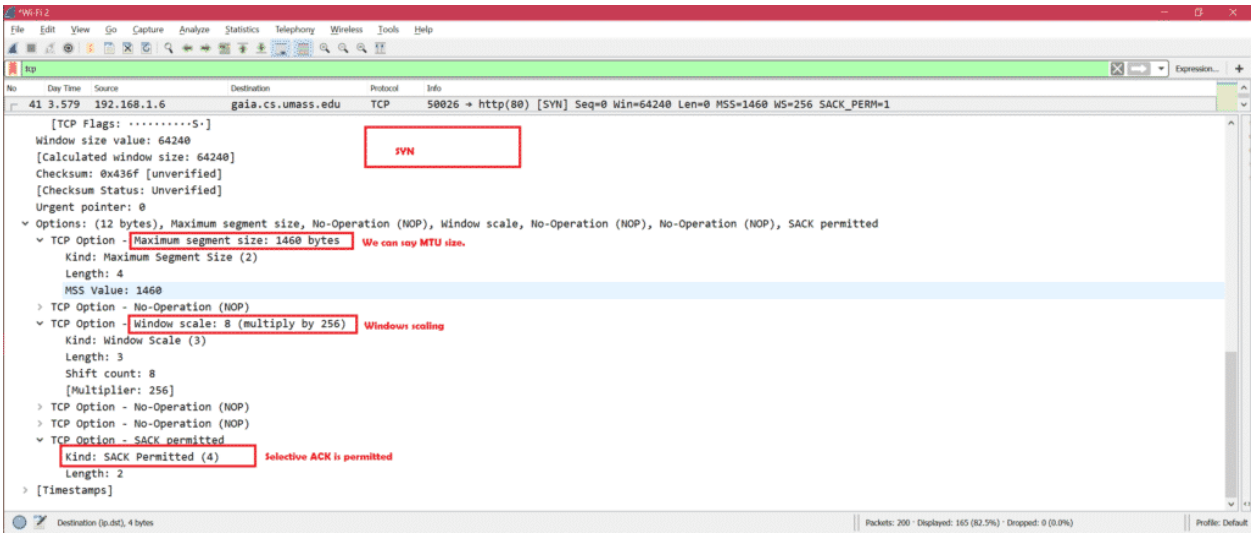


Frame 1 : SYN [Synchronaziation]

SYN is the first packet comes from the client to server. In our case 192.168.1.6 is the client [The system where we opened the browser] and gaia.cs.umass.edu is the server.

Here are some important fields in SYN frame



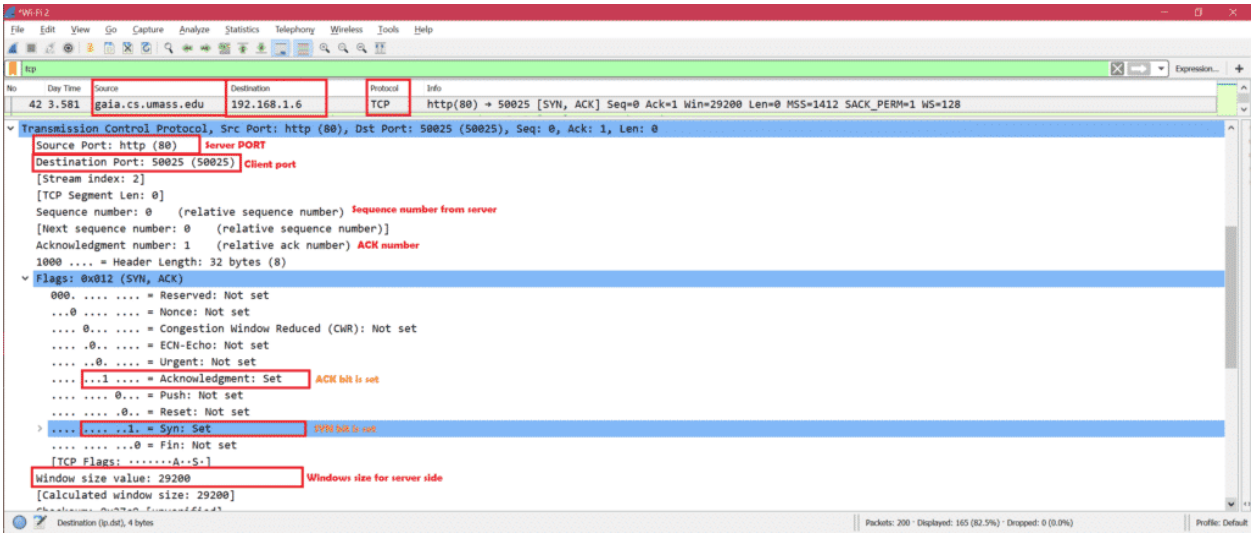


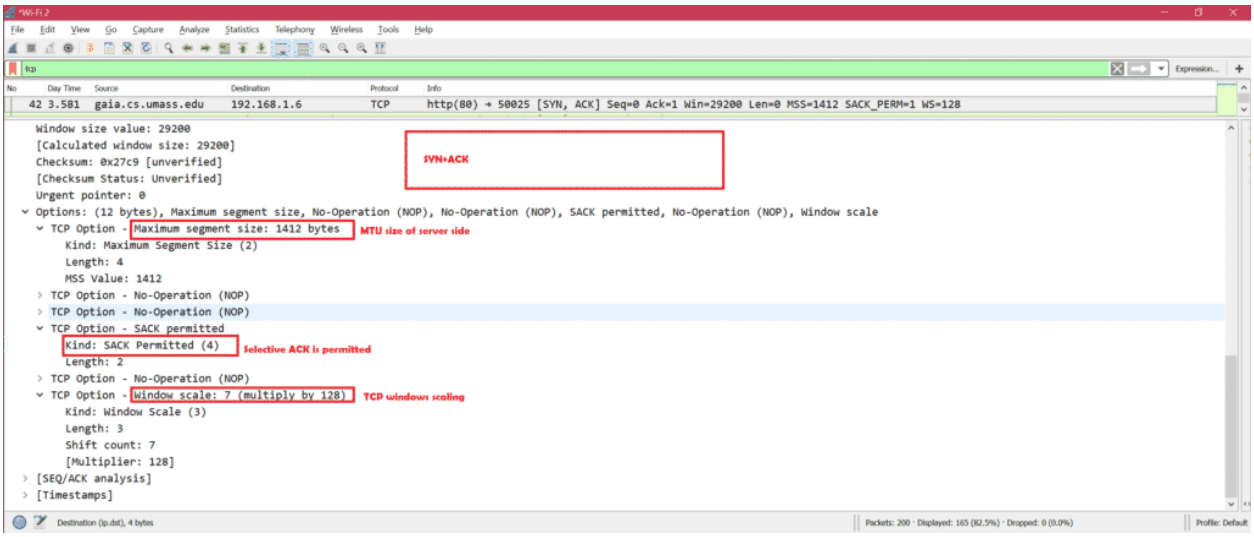
SYN frame is required to send the capabilities of client to server.

Frame 2 : SYN+ACK [Synchronaziation + Acknowledgement]

SYN, ACK is the second packet comes from the server to client.

Here are some important fields in SYN, ACK frame





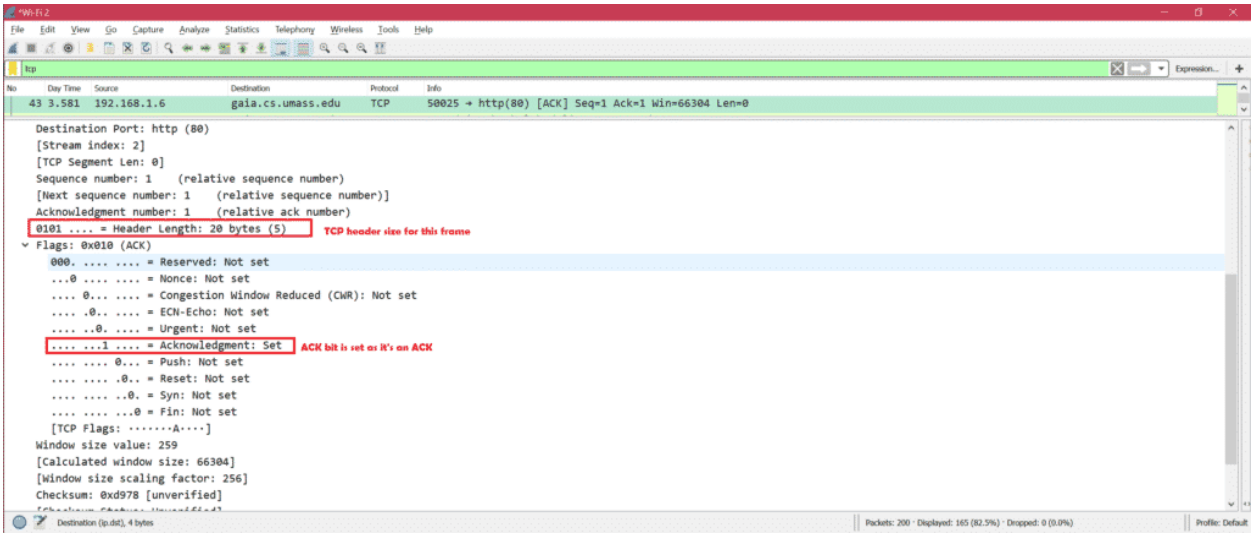
SYN, ACK frame is required to send the capabilities of server to client.

Now client and server have shared their capabilities.

Frame 3 : ACK [Acknowledgement]

ACK is the third packet comes from the client to server. This is basically an acknowledgement from client to server and also it's an acceptance of capabilities sent by server.

Here are the important fields for ACK.



Let's check the important informations shared between client and server:

Client	**Server**
Receive Window Size: **64240 Bytes**	Receive Window Size: **29200 Bytes**
Maximum segment size: **1460 bytes**	Maximum segment size: **1412 bytes**
SACK Permitted: **Yes**	SACK Permitted: **Yes**
Window scale: **8 (multiply by 256)**	Window scale: **7 (multiply by 128)**

We have noticed there are differences in values. If client or server accepts other's capabilities then 3-way handshake is successful.

TCP Header:

Here are the important fields of TCP header:

1. **Source port (16 bits):** This is the sending port.

Example: Source Port: 50026 (50026)

2. **Destination port (16 bits):** This is the receiving port.

Example: Destination Port: http (80)

3. **Sequence number (32 bits):**

- If SYN bit is set [1] then this is initial sequence number.
- If SYN bit is not set [0] then this is the accumulated sequence number of the first data byte of this segment.

Example: Sequence number: 0 (relative sequence number)

4. **Acknowledgment number (32 bits):** If the ACK flag is set then the value of this field is the next sequence number that the sender of the ACK is expecting.

Example: Acknowledgment number: 0

5. **Header Length:** Header size may vary from 20 bytes and maximum of 60 bytes.

Example: 1000 .... = Header Length: 32 bytes (8)

6. **Flags (9 bits):**

Example:
.... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... .... = Congestion Window Reduced (CWR): Not set
.... .0.. .... = ECN-Echo: Not set
.... ..0. .... = Urgent: Not set
....0 .... = Acknowledgment: Not set
.... 0... = Push: Not set
.... .0.. = Reset: Not set
....1. = Syn: Set
....0 = Fin: Not set

7. **Window size (16 bits):** This is the size of receive window in bytes.

Example: Window size value: 64240

8. **Checksum (16 bits):**

It's used error-checking of the header.

Example: Checksum: 0x436f

9. **Urgent pointer (16 bits):**

This is an offset from the sequence number indicating the last urgent data byte.

Example : Urgent pointer: 0

10. **Options:**

Example:

TCP Option - Maximum segment size: 1460 bytes
TCP Option - No-Operation (NOP)
TCP Option - Window scale: 8 (multiply by 256)
TCP Option - SACK permitted

Observation:

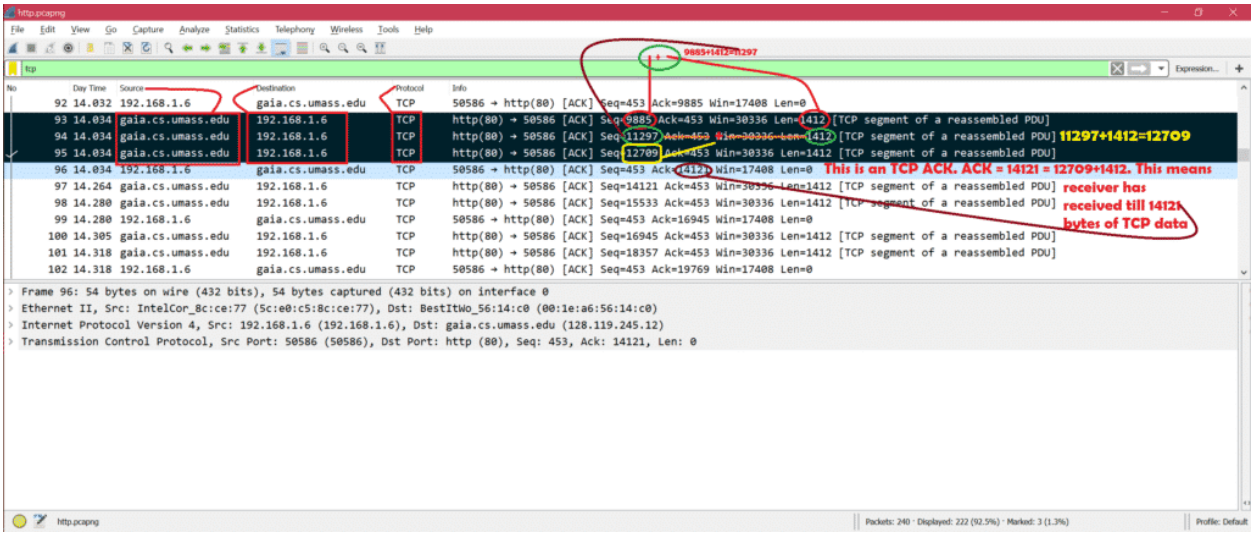
TCP Header size of SYN is 32 Bytes.

TCP Header size of SYN, ACK is 32 Bytes.

TCP Header size of ACK is 20 Bytes as it does not have option fields.

TCP Data:

Here is the screenshot with explanation for TCP data and TCP ACK. Here we can see TCP delay ACK feature. Server has sent three TCP data packets to client and client has sent one delay ACK to tell server that it has received all three TCP data packets. That's why in TCP ACK [Packet number 96 in screenshot] we see ACK=14121 which means client has received till 14121 bytes.



Reference:

For basic theory of TCP refer