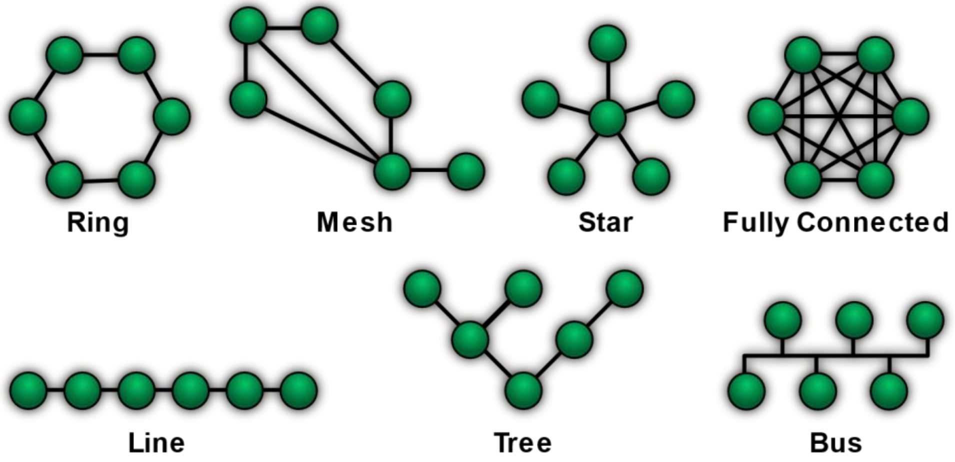
Network

- A network, in computing, is a group of two or more devices or nodes that can communicate. The devices or nodes in question can be connected by physical or wireless connections. The key is that there are at least two separate components, and they are connected.
- Topology is the arrangement of the elements of a communication network. Network topology can be used to define or describe the arrangement of various types of telecommunication networks, including command and control radio networks and computer networks.
- This network has a universal language called osi(Open Systems Interconnection), which all systems that want to exist in the network must comply with this model in order to be able to communicate.



OSI Model

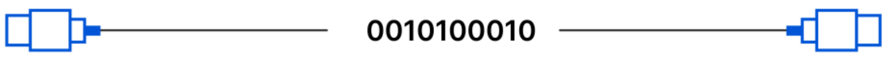
- The OSI Model can be seen as a universal language for computer networking. It is based on the concept of splitting up a communication system into seven abstract layers, each one stacked upon the last.
- This model is introduced with seven layers, each layer does not conflict with the lower layer. This means that the higher layers do not need to understand the logic of the lower layers, for example, a simple request does not matter if it comes from a Linux or Android system.If they are in the same network, they can see each other and communicate with each other
- The Open Systems Interconnection (OSI) model is a way to represent how devices communicate with one another. It consists of seven layers:
- You receive data from layers 1 through 7 and transmit data in the opposite direction. That's because every layer of the OSI Model handles a specific job and passes data to and from the layers above and below itself.
- When the information is sent from layer one to layer seven, headers and flags related to that layer are added to the packet in each layer to finally become something that the application understands.

1. Physical
2. Data link
3. Network
4. Transport
5. Session
6. Presentation
7. Application

1. The physical layer

- This layer includes the physical equipment involved in the data transfer, such as the cables and switches. This is also the layer where the data gets converted into a bit stream, which is a string of 1s and 0s. The physical layer of both devices must also agree on a signal convention so that the 1s can be distinguished from the 0s on both devices.

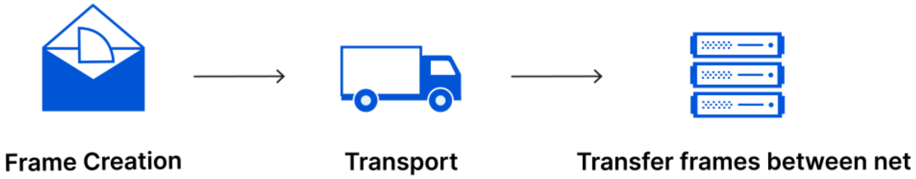
The Physical Layer



2. The data link layer

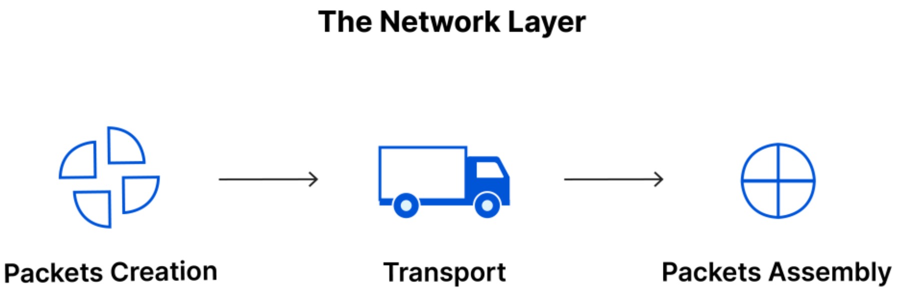
- The data link layer is very similar to the network layer, except the data link layer facilitates data transfer between two devices on the *same* network. The data link layer takes packets from the network layer and breaks them into smaller pieces called frames. Like the network layer, the data link layer is also responsible for flow control and error control in intra-network communication (The transport layer only does flow control and error control for inter-network communications).

The Data Link Layer



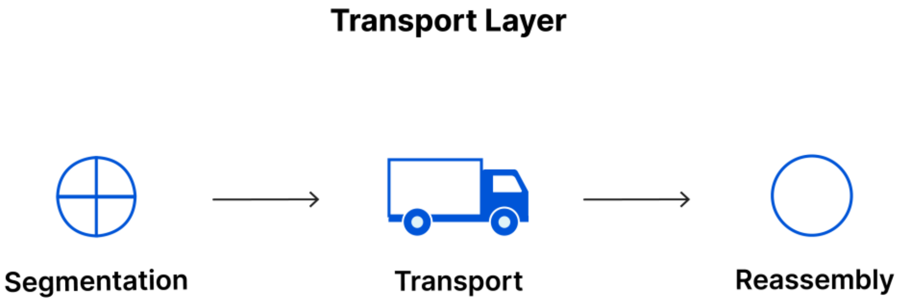
3. The network layer

- The network layer is responsible for facilitating data transfer between two different networks. If the two devices communicating are on the same network, then the network layer is unnecessary. The network layer breaks up segments from the transport layer into smaller units, called packets, on the sender's device, and reassembling these packets on the receiving device. The network layer also finds the best physical path for the data to reach its destination; this is known as routing.
- Network layer protocols include IP, the Internet Control Message Protocol (ICMP), the Internet Group Message Protocol (IGMP), and the IPsec suite.



4. The transport layer

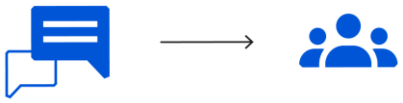
- Layer 4 is responsible for end-to-end communication between the two devices. This includes taking data from the session layer and breaking it up into chunks called segments before sending it to layer 3. The transport layer on the receiving device is responsible for reassembling the segments into data the session layer can consume
- Transport layer protocols include the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).



5. The session layer

- This is the layer responsible for opening and closing communication between the two devices. The time between when the communication is opened and closed is known as the session. The session layer ensures that the session stays open long enough to transfer all the data being exchanged, and then promptly closes the session in order to avoid wasting resources.

The Session Layer



Session of communication

6. The presentation layer

This layer is primarily responsible for preparing data so that it can be used by the application layer; in other words, layer 6 makes the data presentable for applications to consume. The presentation layer is responsible for translation, encryption, and compression of data.

The Presentation Layer

Encryption

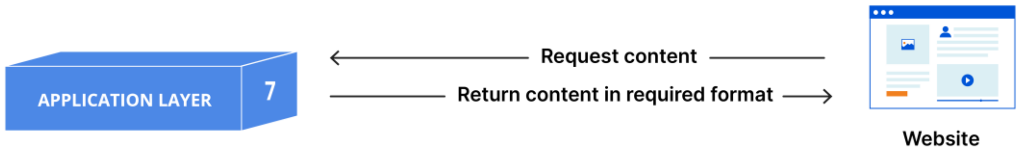
Compression

Translation

7. The application layer

This is the only layer that directly interacts with data from the user. Software applications like web browsers and email clients rely on the application layer to initiate communications. But it should be made clear that client software applications are not part of the application layer; rather the application layer is responsible for the protocols and data manipulation that the software relies on to present meaningful data to the user. Application layer protocols include HTTP as well as SMTP (Simple Mail Transfer Protocol is one of the protocols that enables email communications).

Application Layer



APPLICATION LAYER

7

Request content

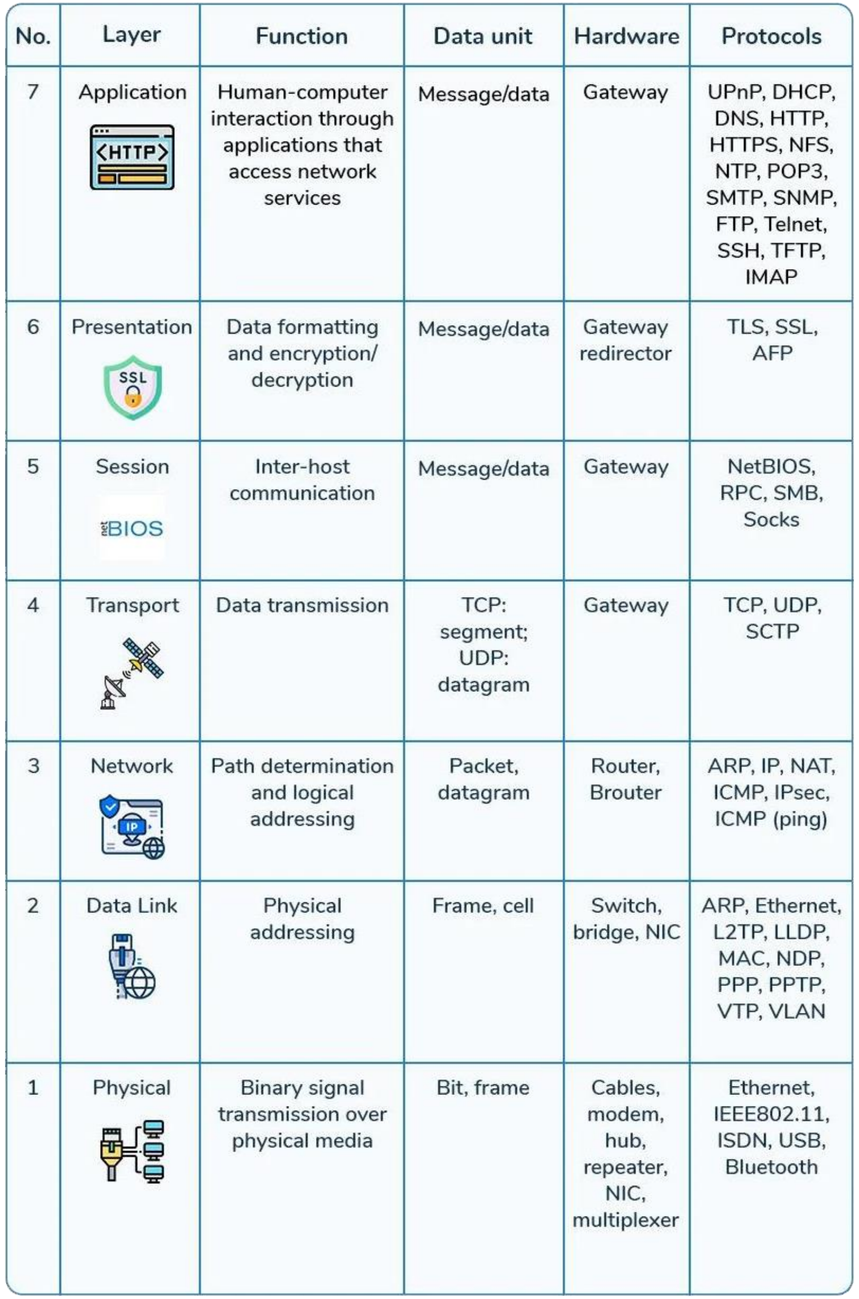
Return content in required format

Website

All OSI Layers Summery:

- One of the most important protocols mentioned is TCP, which exists in layer four
- There is a famous term called tcp hanshake, which we will explain in the next slide

- information in tcp according to mss (maximum segment size)They fall into pieces,The sender converts the information into segments and the receiver reassembles the segments We Call This **TCP REASSEMBLY**



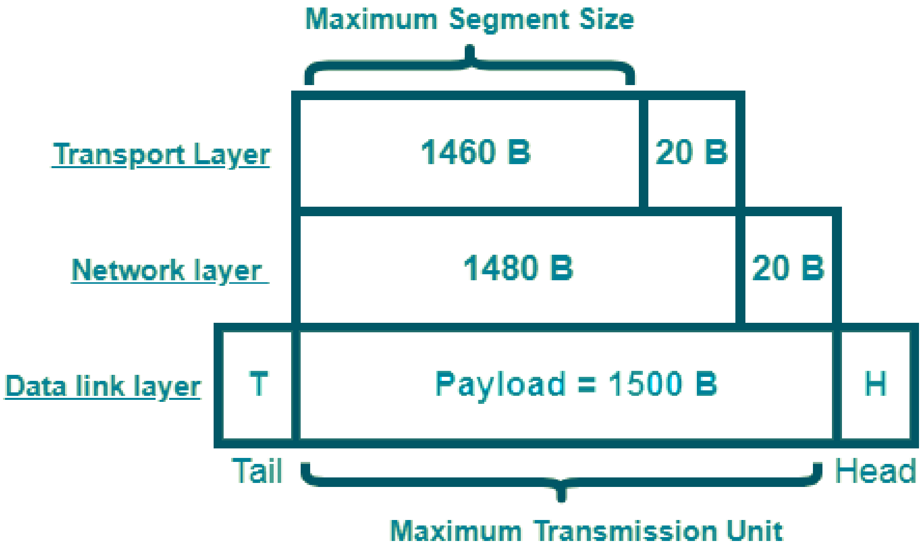
No.	Layer	Function	Data unit	Hardware	Protocols
7	Application	Human-computer interaction through applications that access network services	Message/data	Gateway	UPnP, DHCP, DNS, HTTP, HTTPS, NFS, NTP, POP3, SMTP, SNMP, FTP, Telnet, SSH, TFTP, IMAP
6	Presentation	Data formatting and encryption/ decryption	Message/data	Gateway redirector	TLS, SSL, AFP
5	Session	Inter-host communication	Message/data	Gateway	NetBIOS, RPC, SMB, Socks
4	Transport	Data transmission	TCP: segment; UDP: datagram	Gateway	TCP, UDP, SCTP
3	Network	Path determination and logical addressing	Packet, datagram	Router, Brouter	ARP, IP, NAT, ICMP, IPsec, ICMP (ping)
2	Data Link	Physical addressing	Frame, cell	Switch, bridge, NIC	ARP, Ethernet, L2TP, LLDP, MAC, NDP, PPP, PPTP, VTP, VLAN
1	Physical	Binary signal transmission over physical media	Bit, frame	Cables, modem, hub, repeater, NIC, multiplexer	Ethernet, IEEE802.11, ISDN, USB, Bluetooth

TCP

- **TCP (Transmission Control Protocol)** is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing reliable delivery services.
- The position of TCP is at the transport layer (Layer 4) of the OSI model. TCP also helps in ensuring that information is transmitted accurately by establishing a virtual connection between the sender and receiver.
- To make sure that each message reaches its target location intact, the TCP/IP model breaks down the data into small bundles and afterward reassembles the bundles into the original message on the opposite end. Sending the information in little bundles of information makes it simpler to maintain efficiency as opposed to sending everything in one go.
- Now suppose we want to download a photo.The size of this photo is so large that it cannot fit in a packet (this size limit is called MSS).But before the sender converts the photo into smaller segments, they must communicate with the receiverTo establish this communication, the receiver and sender send SYN and ACK to each other, which is called **3-WAY HANDSHAKE**

Maximum Segment Size (MSS)

- The maximum segment size, or MSS, sets a restriction on the size of data packets that can be sent over a network like the Internet. Every bit of data that moves across a network is divided up into packets. Multiple headers, each containing information about the contents and destination, are appended to packets. The payload, or non-header portion of a packet, is measured by MSS.
- MSS is like a scale that measures only the trailer if a data packet is like a transport vehicle, where the payload is the trailer and freight and the header is the truck itself. The truck is not permitted to go to its destination if the trailer weighs too much.



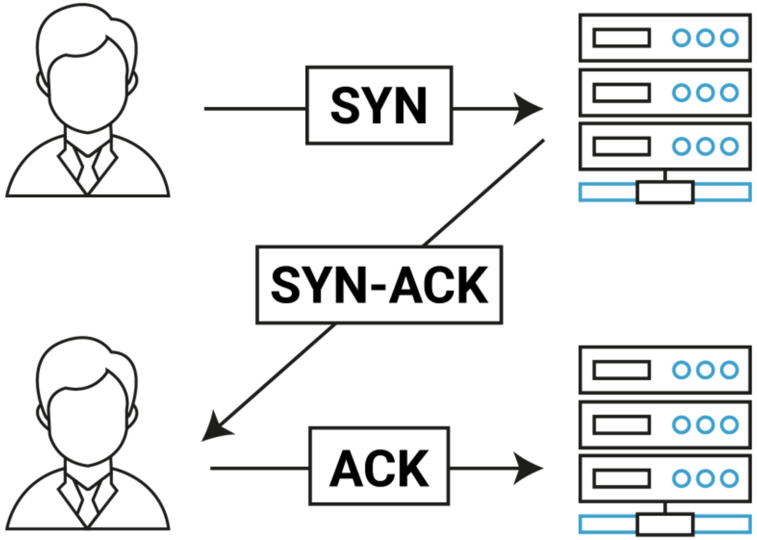
MTU – (IP header + TCP header) = Maximum Segment Size (MSS)

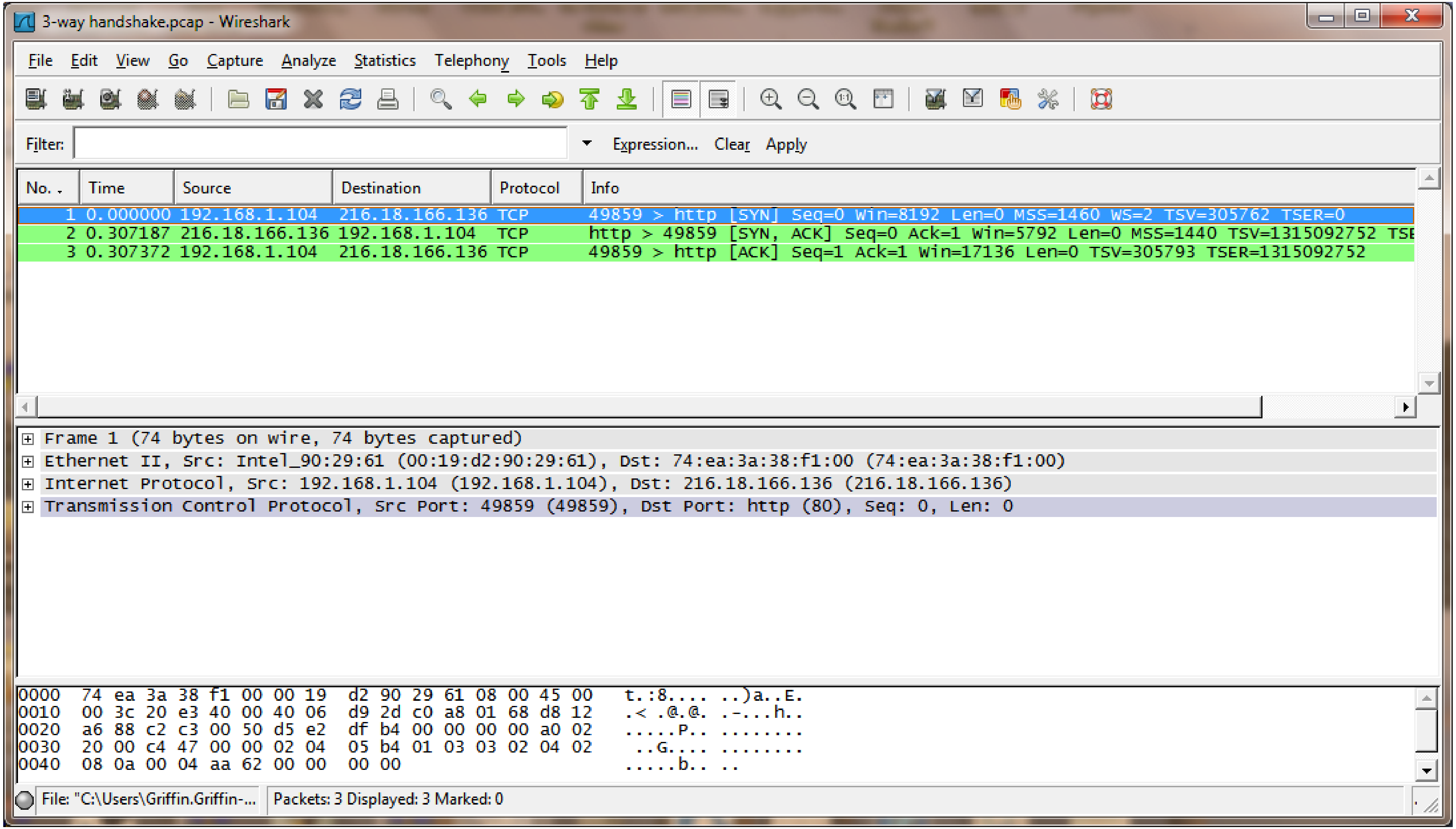
TCP 3-Way Handshake Process

- This could also be seen as a way of how TCP connection is established. Before getting into the details, let us look at some basics. TCP stands for Transmission Control Protocol which indicates that it does something to control the transmission of the data in a reliable way.

- **Step 1 (SYN):** In the first step, the client wants to establish a connection with a server, so it sends a segment with SYN(Synchronize Sequence Number) which informs the server that the client is likely to start communication and with what sequence number it starts segments with
- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of the segment it received and SYN signifies with what sequence number it is likely to start the segments with
- **Step 3 (ACK):** In the final part client acknowledges the response of the server and they both establish a reliable connection with which they will start the actual data transfer

- When the connection is established, the sender immediately sends the segments (fragmented information) and the receiver tells the sender to give it again after receiving each segment, and finally the sender sends the last segment with the FIN flag (this means that the series the information is finished) and finally the receiver puts all these segments together, which is called TCP Reassembly





```
No. .  Time      Source          Destination     Protocol  Info
   1 0.000000 192.168.1.104  216.18.166.136 TCP      49859 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2 TSV=305762 TSER=0
   2 0.307187 216.18.166.136 192.168.1.104  TCP      http > 49859 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1440 TSV=1315092752 TSE
   3 0.307372 192.168.1.104  216.18.166.136 TCP      49859 > http [ACK] Seq=1 Ack=1 Win=17136 Len=0 TSV=305793 TSER=1315092752
```

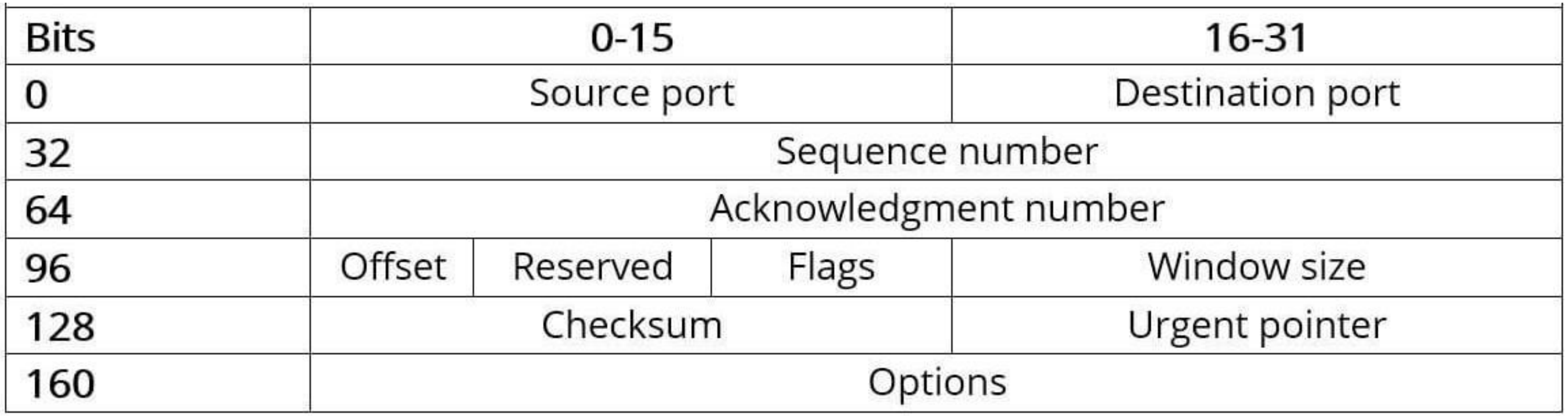
```
⊞ Frame 1 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: Intel_90:29:61 (00:19:d2:90:29:61), Dst: 74:ea:3a:38:f1:00 (74:ea:3a:38:f1:00)
⊞ Internet Protocol, Src: 192.168.1.104 (192.168.1.104), Dst: 216.18.166.136 (216.18.166.136)
⊞ Transmission Control Protocol, Src Port: 49859 (49859), Dst Port: http (80), Seq: 0, Len: 0
```

```
0000  74 ea 3a 38 f1 00 00 19  d2 90 29 61 08 00 45 00   t.:8.... ..)a..E.
0010  00 3c 20 e3 40 00 40 06  d9 2d c0 a8 01 68 d8 12   .< .@.@. .-...h..
0020  a6 88 c2 c3 00 50 d5 e2  df b4 00 00 00 00 a0 02   .....P.. ........
0030  20 00 c4 47 00 00 02 04  05 b4 01 03 03 02 04 02    ..G.... ........
0040  08 0a 00 04 aa 62 00 00  00 00                     .....b.. ..
```

File: "C:\Users\Griffin.Griffin-... Packets: 3 Displayed: 3 Marked: 0

TCP Header

- **Source Port:** Specifies the source port quantity, which identifies the sending utility at the supply tool.
- **Destination Port:** Specifies the destination port wide variety, which identifies the receiving utility on the vacation spot tool.
- **Sequence Number:** Specifies the sequence variety of the first information byte in the TCP section.
- **Acknowledgment Number:** Specifies the subsequent sequence quantity predicted by means of the sender of the TCP phase.
- **Data Offset:** Specifies the period of the TCP header in 32-bit phrases.
- **Reserved**: Reserved for future use and need to be set to zero.
- **Flags:** Various flags that manipulate the behavior of the TCP segment, consisting of SYN (synchronize), ACK (acknowledge), FIN (finish), RST (reset), and others.
- **Window Size:** Specifies the size of the get hold of window, which shows the amount of records that may be received before requiring acknowledgment.
- **Checksum:** Used for errors detection to make sure the integrity of the TCP section in the course of transmission.
- **Urgent Pointer:** Specifies the offset from the series wide variety indicating the end of pressing statistics within the TCP segment.
- **Options:** Optional fields which could encompass extra control records or parameters.



<table>
<tr><th>Bits</th><th colspan="3">0-15</th><th>16-31</th></tr>
<tr><td>0</td><td colspan="3">Source port</td><td>Destination port</td></tr>
<tr><td>32</td><td colspan="4">Sequence number</td></tr>
<tr><td>64</td><td colspan="4">Acknowledgment number</td></tr>
<tr><td>96</td><td>Offset</td><td>Reserved</td><td>Flags</td><td>Window size</td></tr>
<tr><td>128</td><td colspan="3">Checksum</td><td>Urgent pointer</td></tr>
<tr><td>160</td><td colspan="4">Options</td></tr>
</table>

References

My Github Repo To access This network article: https://github.com/MSaLeHNYM/network-article

- www.cloudflare.com/learning
- www.geeksforgeeks.org
- wiki.wireshark.org