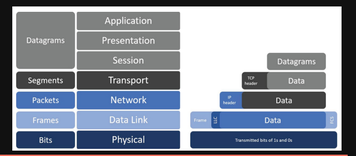
# Encapsulation and Decapsulation




## Encapsulation & Decapsulation

### Encapsulation

Every layer has identifying information such as a port number, the TCP or UDP protocol, IP address, and a frame.

As data is transmitted, data encapsulation is the process of passing a PDU down to the next layer in the protocol stack.

This information explains to the current layer which upper layer the payload (data) came from; this will be important for the decapsulation process.

### Decapsulation

As data is received, data decapsulation is the process of passing the payload (data) up to the next layer in the protocol stack.

When the payload is decapsulated, the information is read from the PDU header or type field (frame).

This allows the current layer to know which upper layer to pass the payload to.

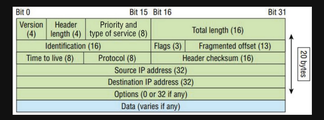
## IP

The Internet Protocol (IP) is a Network layer protocol that allows for the logical addressing of networks and hosts.

UDP and TCP function on top of the IP protocol.

UDP and TCP are protocols that handle the data for the applications.

The IP protocol is responsible for encapsulating these protocols and delivering it to the appropriate addresses.




IP Packet

## MTU

The maximum transmission unit (MTU) is the largest size of the data that can be transferred at the Data Link layer.

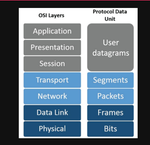
The MTU for Ethernet is 1500 bytes.

Adding 12 bytes for the destination and source MAC address, a 2-byte type field, and 4 bytes for the frame check sequence (FCS) brings the MTU to 1518 bytes.

The smallest MTU is 46 bytes, or 64 bytes if including the frame fields.

The MTU is often referred to as a layer 3 data size. When data is passed down to the Data Link layer, the packet is sized to the MTU of the Data Link layer.

The Ethernet specification allows for either an MTU of 1500 bytes or an MTU of 9000. When the MTU is increased to 9000 bytes, the frame is considered a jumbo frame.




Protocol Data Units (PDU)

The term protocol data units (PDUs) is how we describe the type of data transferred at each layer of the OSI model.

The PDU for these upper layers is considered user datagrams, or just datagrams. The datagrams are created by the application and passed to the Transport layer.

The Transport layer is where segments are created from the datagrams, and then the segments are passed to the Network layer.

At the network layer, packets are created from the segments and passed to the Data Link layer.

The Data Link layer creates frames for transmitting the data in bits at the Physical layer.

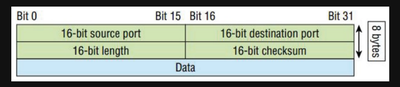
## UDP

The User Datagram Protocol (UDP) is a transport protocol for TCP/IP.

UDP is connectionless, which means that data is simply passed from one IP address over the network to the other IP address.

The receipt of the data is not acknowledged by the destination computer.

The sending computer won't know if the destination computer is even listening.




UDP Segment

UDP is used because it is faster than TCP.

## TCP

Transmission Control Protocol (TCP) is another transport protocol for TCP/IP.

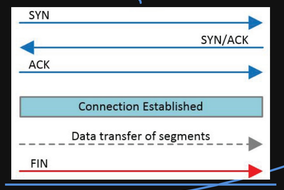
TCP is a connection-oriented protocol. During the transmission of information, both ends create a virtual circuit over the network.

When the sending computer transmits data to a receiving computer, a virtual connection is created using a three-way handshake.

During the three-way handshake, the window buffer size on each side is negotiated with the SYN and ACK flags in the TCP header.

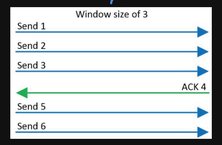
When both the sender and receiver acknowledge the window's size, the connection is considered established and data can be transferred.

When the data transfer is completed, the sender can issue a FIN flag in the TCP header to tear down the virtual connection.




### TCP Sliding Window

The buffer size negotiated in the three-way handshake determines the sliding TCP window for acknowledgment of segments during the transfer.



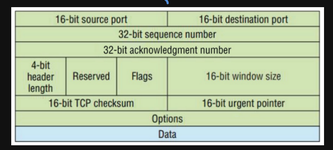

An example of TCP sliding window size 3.

After three sequenced packets are delivered and put back in order on the receiving computer, the receiving computer sends an acknowledgment for the next segment it expects.

### Lost Segment

The acknowledge timer will be triggered on the receiver and the receiver will acknowledge the segments it currently has received.

The sender's retransmit timer will also expire, and the lost segment will be retransmitted to the receiving computer.




TCP Segment