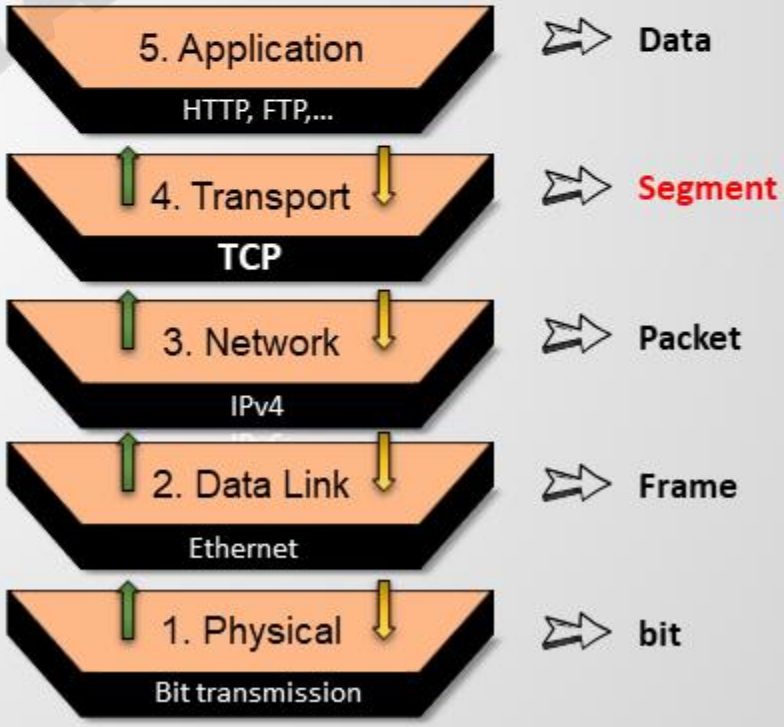
Transmission Control Protocol (TCP)

- **The Transmission Control Protocol (TCP)** is one of the main protocols of the Internet protocol suite, It complements the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP.

- **TCP** provides **error recovery** (reliability) and **flow control** using **windowing** between applications running on hosts communicating via an IP network.

- **TCP** is connection-oriented, and a connection between client and server is established before data can be sent.

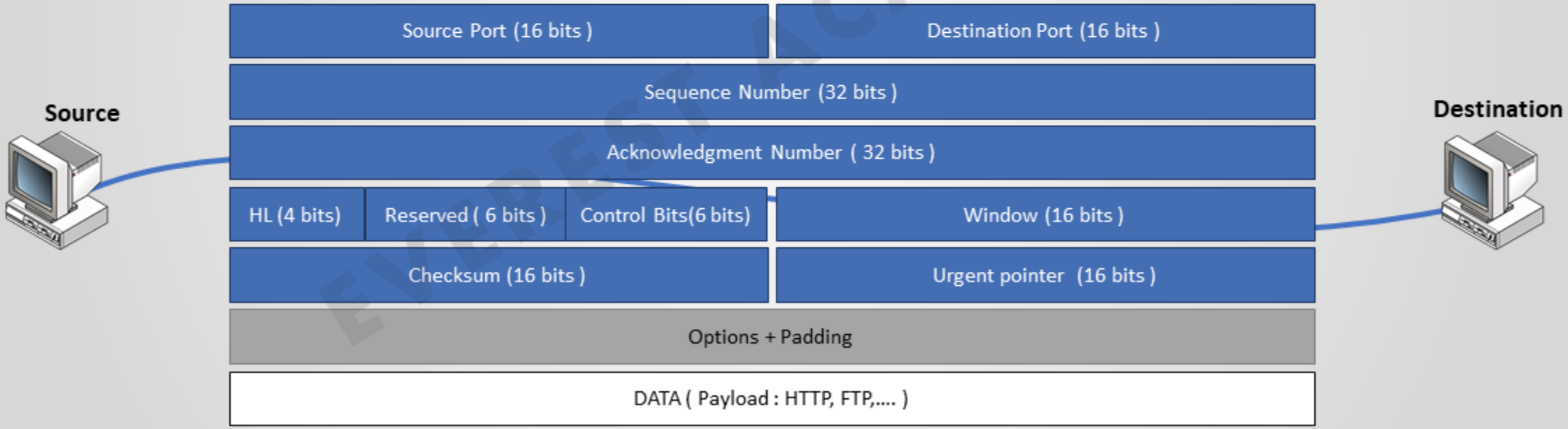
- **Major internet applications** such as the World Wide Web, email, remote administration, and file transfer rely on TCP.



TCP Segment

- **A TCP Segment** is composed of a header and payload. The header consists of fixed and optional fields. The payload appears immediately after the header. A TCP Segment is often carried as the payload inside an IP packet.

16 + 16 + 32 + 32 + 4 + 6 +6 + 16 + 16 + 16 = 160 bits = 20 Bytes or Octets

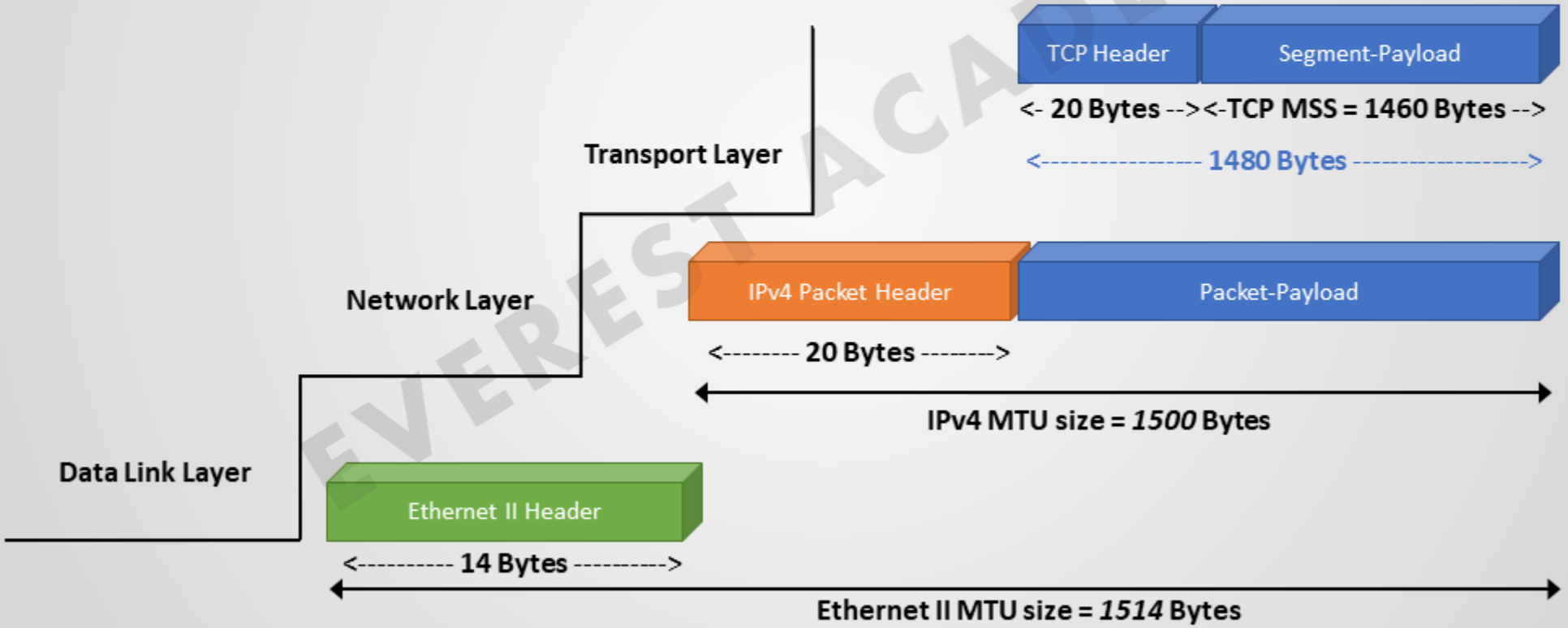


TCP Segment

Source Port (16 bits)	The TCP Source Port is the port number used by the computer sending the TCP segment
Destination Port (16 bits)	The TCP Destination Port is the port number used by the computer receiving the TCP segment
Sequence Number (32 bits)	The sequence number of the first data Byte in this segment.
Acknowledgment Number (32 bits)	The acknowledgement number is the sequence number of the **next** byte the receiver expects to receive.
Header Length (4 bits)	It indicates the number of bytes in the TCP header and allows the receiver to jump directly to the data.
Reserved (6 bits)	Reserved (6 bits)
Control Bits (6 bits)	URG : Urgent Pointer field, ACK : Acknowledgment field, PSH : Push Function, RST : Reset the connection, SYN : Synchronize sequence numbers, FIN : No more data from sender
Window (16 bits)	It specifies the number of Bytes the receiver is willing to receive
Checksum (16 bits)	It is used for integrity checks TCP segment
Urgent pointer (16 bits)	It indicates the end of urgent data.

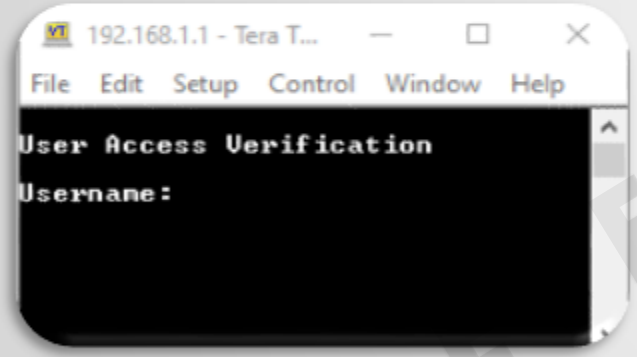
Maximum Segment Size (MSS)

- **The maximum segment size (MSS)** is a parameter of the options field of the TCP header that specifies the largest amount of data, specified in bytes, that a computer or communications device can receive in a single TCP segment.

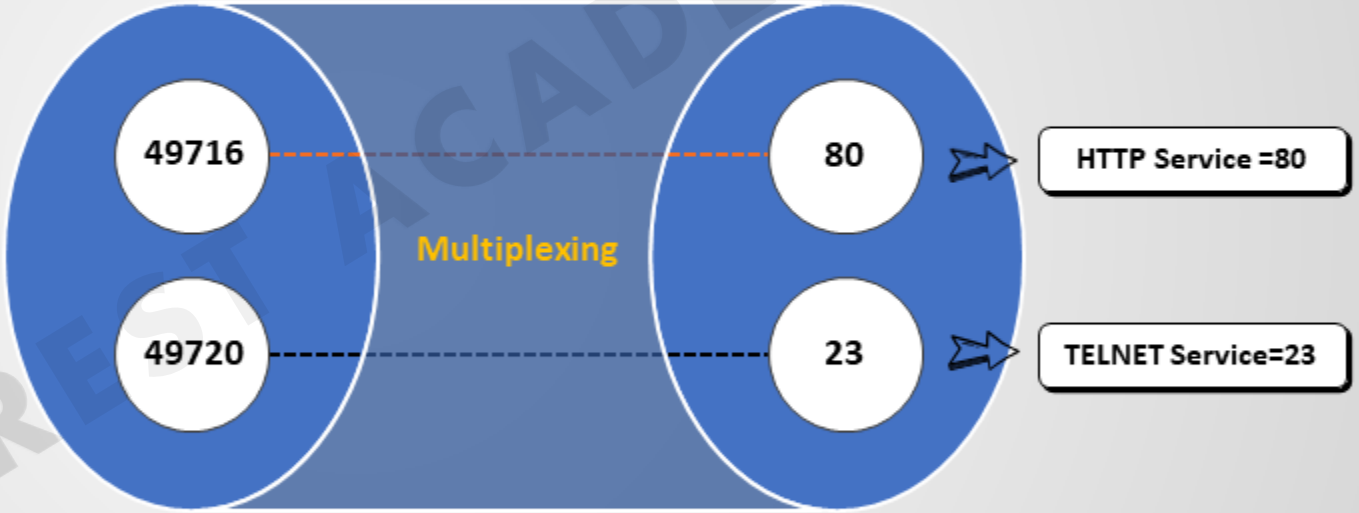


Port Numbers and Multiplexing

HTTP Port = 49716



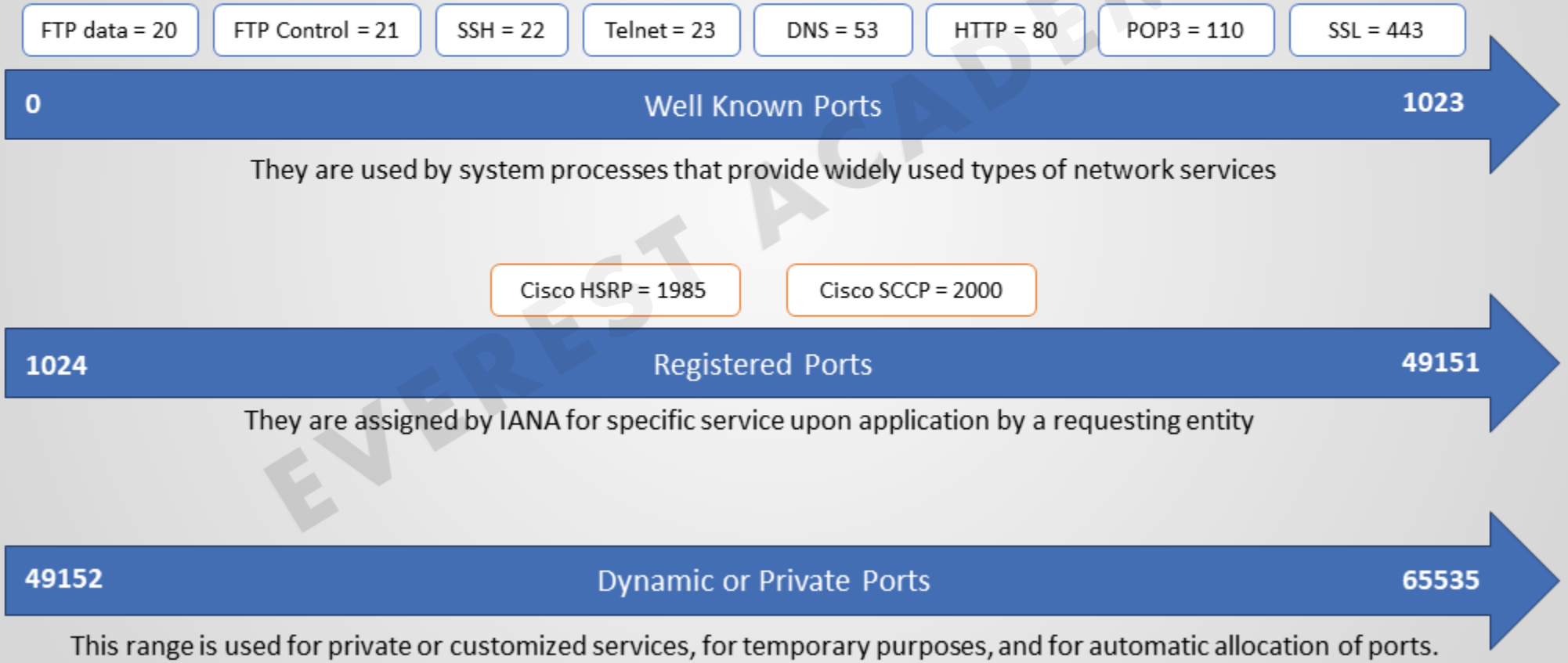
TELNET Port = 49720



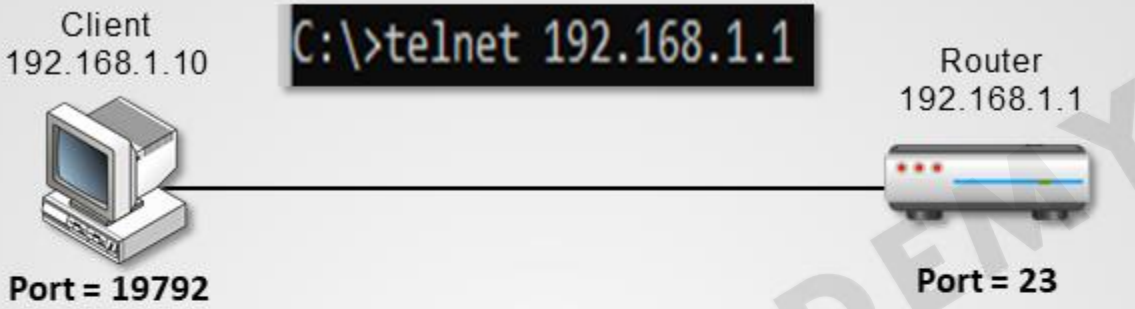

```
C:\>netstat -n

Active Connections

  Proto  Local Address          Foreign Address
  TCP    192.168.1.10:49716     192.168.1.1:23
  TCP    192.168.1.10:49720     192.168.1.1:80
```

Ranges of Port Numbers (Transport Layer)

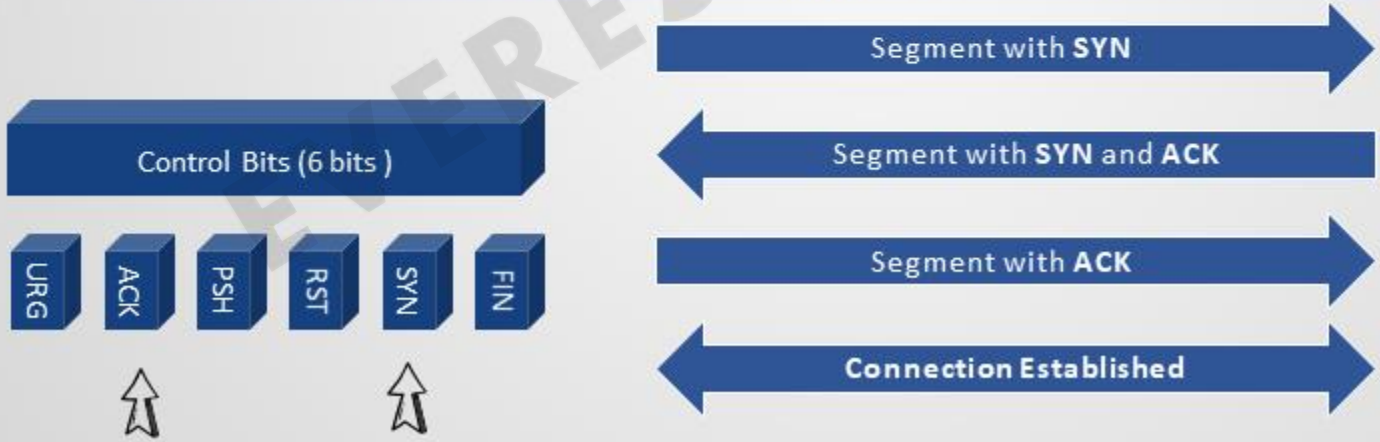
- **The Internet Assigned Numbers Authority (IANA)** subdivides the port number ranges into three main ranges.



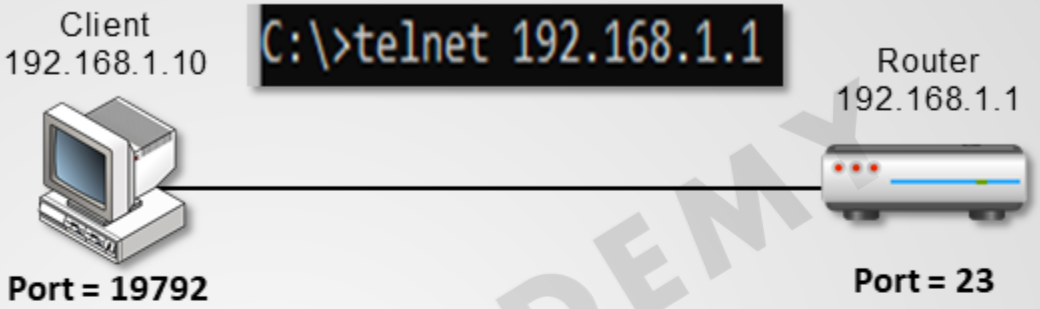
TCP 3-Way Handshake



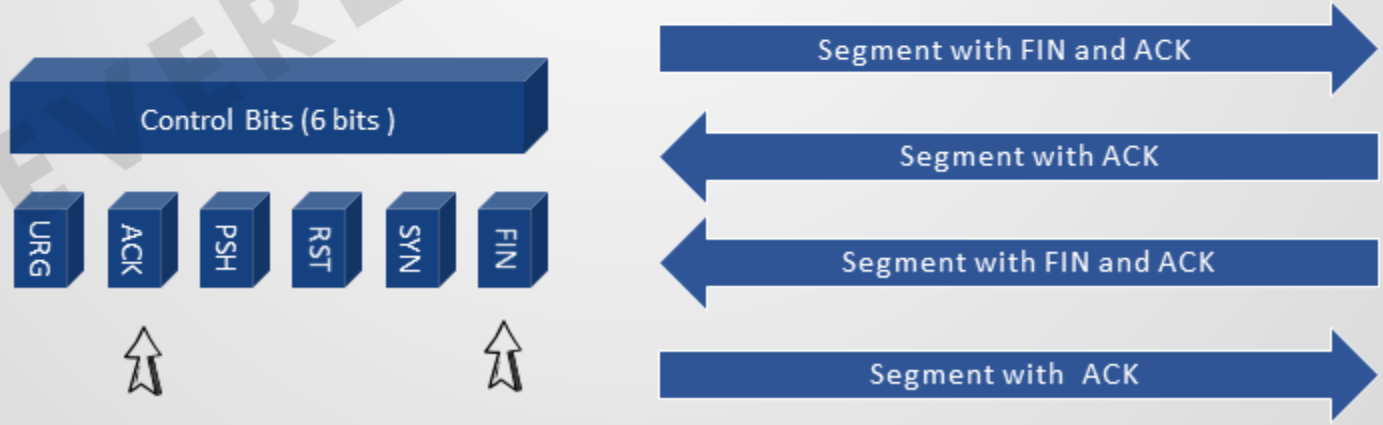
	Source	Destination	Protocol	Info
Can You Hear Me ?	192.168.1.10	192.168.1.1	TCP	19792 → 23 [SYN] Seq=0
Yes, Can You Hear Me ?	192.168.1.1	192.168.1.10	TCP	23 → 19792 [SYN, ACK] Seq=0 Ack=1
Yes.	192.168.1.10	192.168.1.1	TCP	19792 → 23 [ACK] Seq=1 Ack=1
	192.168.1.1	192.168.1.10	TELN…	Telnet Data ...



TCP Connection Termination



	Source	Destination	Protocol	Info
I want to close the Connection, Go Ahead	192.168.1.10	192.168.1.1	TCP	19792 → 23 [FIN, ACK] Seq=51 Ack=91
Okay	192.168.1.1	192.168.1.10	TCP	23 → 19792 [ACK] Seq=91 Ack=52
I want to close the Connection, Go Ahead	192.168.1.1	192.168.1.10	TCP	23 → 19792 [FIN, ACK] Seq=91 Ack=52
Okay	192.168.1.10	192.168.1.1	TCP	19792 → 23 [ACK] Seq=52 Ack=92



TCP Sequence Numbers and Acknowledgments

Sequence Number	Next Sequence Number	Acknowledgment Number	Data

←

Sequence Number	Next Sequence Number	Acknowledgment Number	Data

→

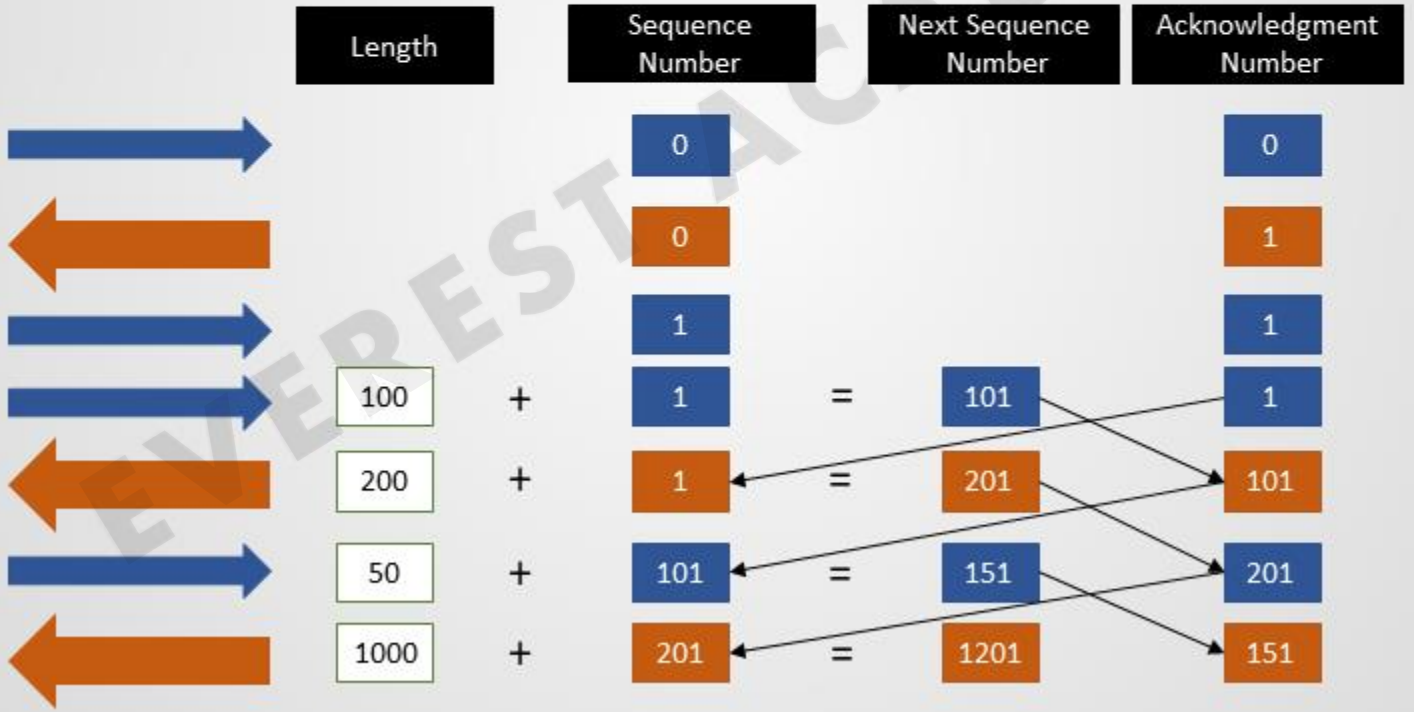
Sequence Number	Next Sequence Number	Acknowledgment Number	Data

←

Sequence Number	Next Sequence Number	Acknowledgment Number	Data

→

TCP Sequence Numbers and Acknowledgments



TCP Flow Control Using Windowing

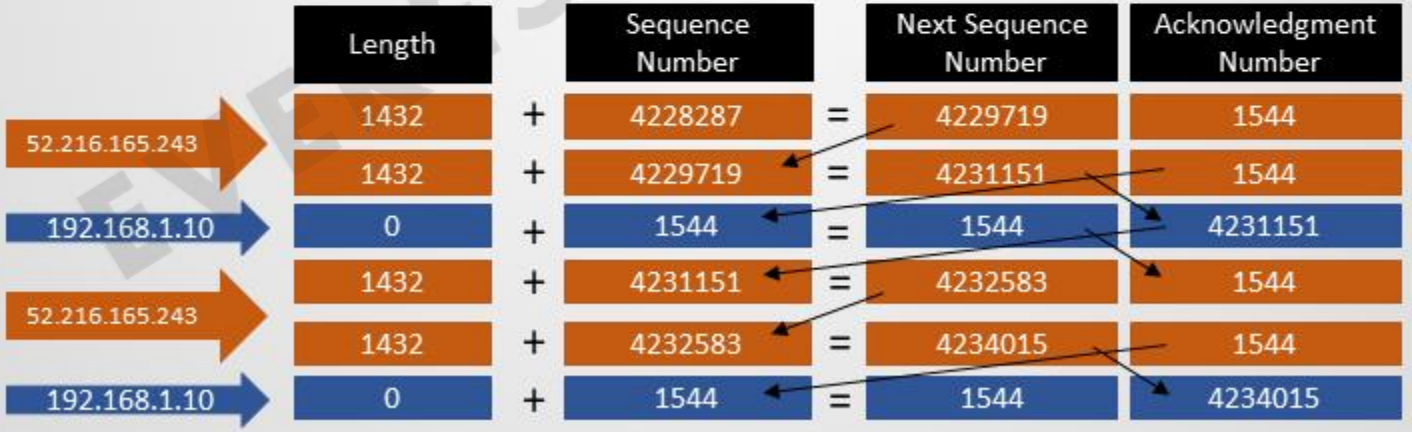
Receive 1	←	Send 1	Receive 1	←	Send 1
Receive 2	←	Send 2	Receive 2	←	Send 2
Receive 3	←	Send 3	Receive 3	← X	Send 3
ACK 4	→		ACK 3	→	
Receive 4	←	Send 4	Receive 3	←	Send 3
Receive 5	←	Send 5	Receive 4	←	Send 4
Receive 6	←	Send 6	Receive 5	← X	Send 5
ACK 7	→		ACK 5	→	
Receive 7	←	Send 7	Receive 5	←	Send 5
Receive 8	←	Send 8	Receive 6	←	Send 6
Receive 9	←	Send 9	ACK 7	→	
ACK 10	→				

TCP Flow Control Using Windowing

	Length		Sequence Number		Next Sequence Number	Acknowledgment Number
→	100	+	1	=	101	1
→	100	+	101	=	201	1
←	0	+	1	=	1	201
→	100	+	201	=	301	1
→	100	+	301	=	401	1
←	0	+	1	=	1	401

TCP Flow Control Using Windowing

Source	Destination	Protocol	Len	Sequence number	Next sequence number	Acknowledgment number	Info
52.216.165.243	192.168.1.10	TCP	1432	4228287	4229719	1544	443 → 18933 [ACK]
52.216.165.243	192.168.1.10	TCP	1432	4229719	4231151	1544	443 → 18933 [ACK]
192.168.1.10	52.216.165.243	TCP	0	1544	1544	4231151	18933 → 443 [ACK]
52.216.165.243	192.168.1.10	TCP	1432	4231151	4232583	1544	443 → 18933 [ACK]
52.216.165.243	192.168.1.10	TCP	1432	4232583	4234015	1544	443 → 18933 [ACK]
192.168.1.10	52.216.165.243	TCP	0	1544	1544	4234015	18933 → 443 [ACK]



TCP Flow Control Using Windowing

Len	Sequence number	Next sequence number	Acknowledgment number
1432	4228287	4229719	1544
1432	4229719	4231151	1544
0	1544	1544	4231151
1432	4231151	4232583	1544
1432	4232583	4234015	1544
0	1544	1544	4234015

SN=4228287, NSN=4229719, ACK= 1544 — 1432

SN=4229719, NSN=4231151, ACK= 1544 — 1432

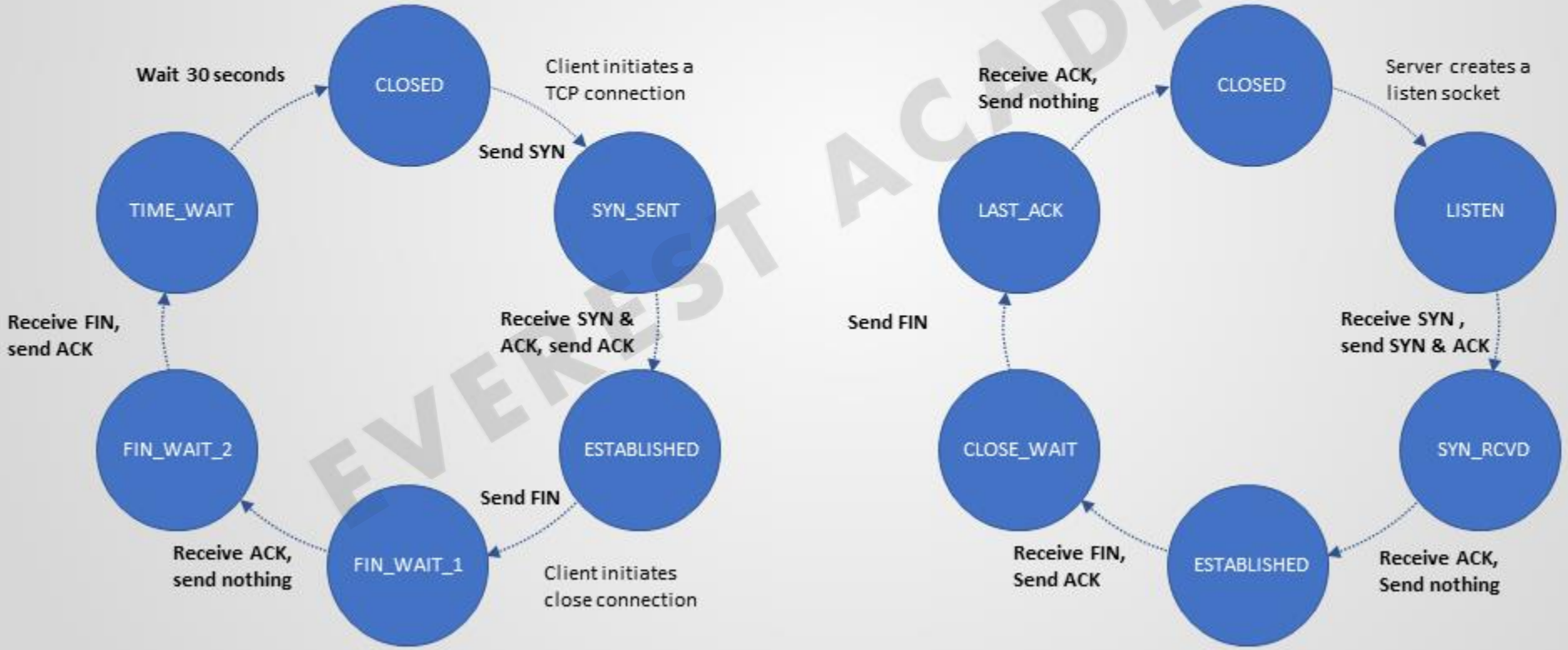
SN=1544, NSN=1544, ACK= 4231151

SN=4231151, NSN=4232583, ACK= 1544 — 1432

SN=4232583, NSN=4234015, ACK= 1544 — 1432

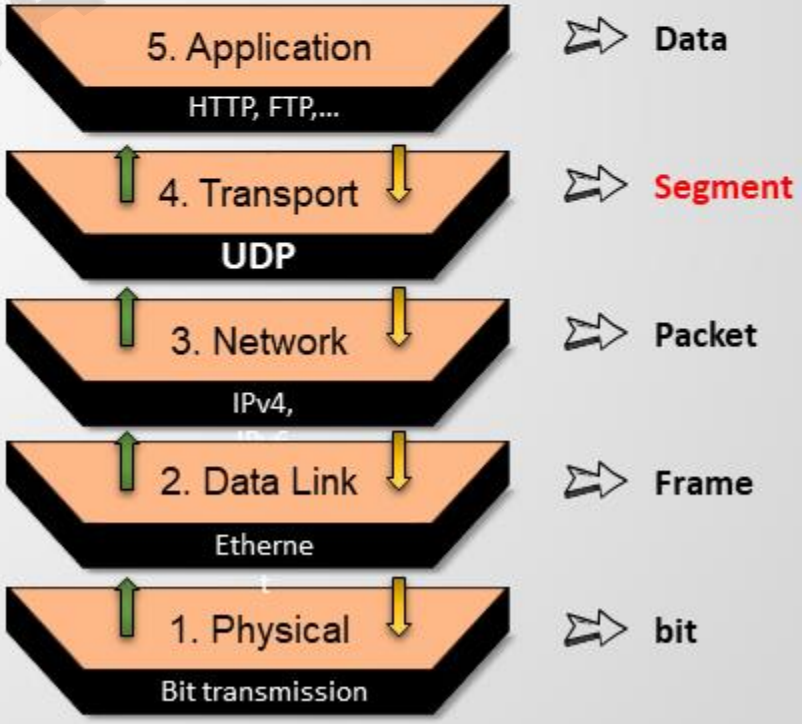
SN=1544, NSN=1544, ACK= 4234015

TCP Connection States



User Datagram Protocol (UDP)

- **User Datagram Protocol (UDP)** is a communication protocol used across the Internet for especially time-sensitive transmissions such as ***video playback*** *or* ***DNS lookups.***
- **UDP** doesn't have the error checking and ordering functionality of TCP and is best utilized when error checking is not needed and speed is important.
- **UDP** doesn't have 3-way handshake to establish a connection between the client and the server as TCP does and hence it is ***unreliable protocol***.
- **Voice** and **video** traffic are sent using UDP because they are both time-sensitive.

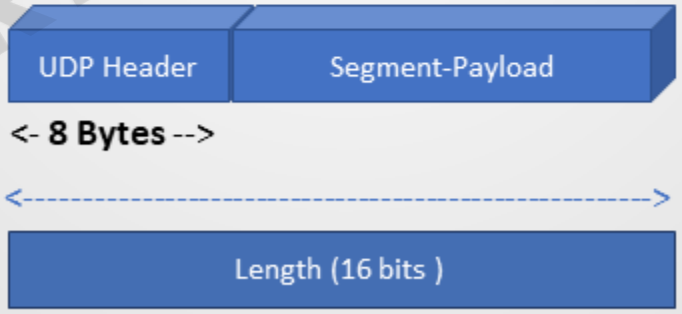


UDP Segment

- **A UDP Segment** is composed of a *header* and *payload*. The payload appears immediately after the header. A UDP Segment is often carried as the payload inside an IP packet.

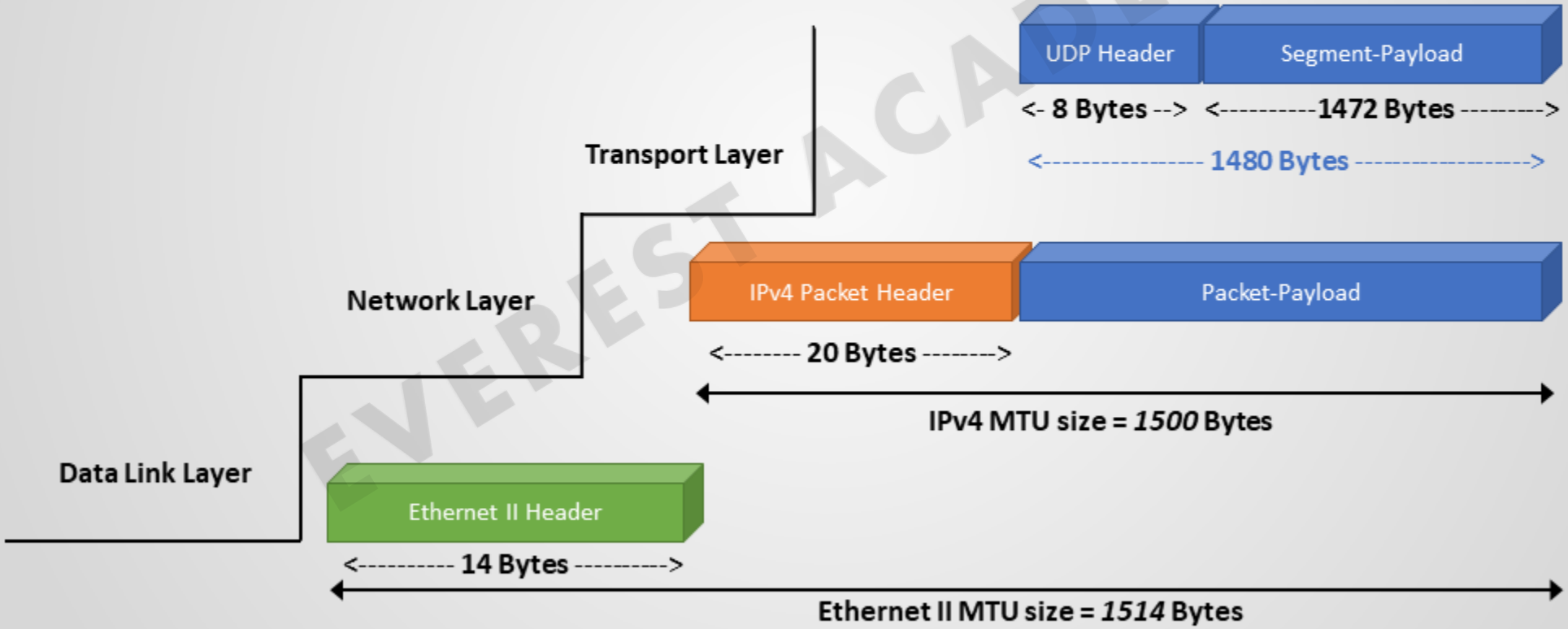
16 + 16 + 16 + 16 = 64 bits = 8 Bytes or Octets

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
DATA (Payload : DNS,....)	

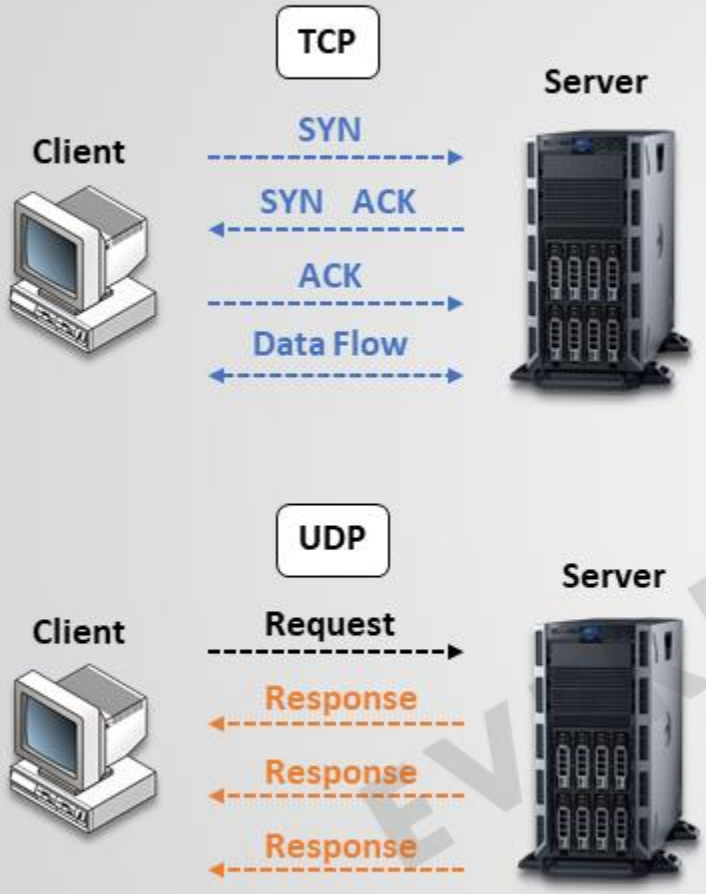


UDP Segment

- **A UDP Segment** is composed of a *header* and *payload*. The payload appears immediately after the header. A UDP Segment is often carried as the payload inside an IP packet.



Differences Between TCP and UDP



TCP	UDP
20 Bytes Header	8 Bytes Header
Connection-Oriented	Connectionless
Flow Control	No Flow Control
Error Checking	No Error Checking
Use Acknowledgment	No Acknowledgement
Three-way Handshake	No Three-way Handshake
Use Sequent Numbers	No Sequent Numbers
Used By Critical Applications	Used By Real-time Applications
HTTP, HTTPs, FTP, Telnet, ...	DNS, DHCP, TFTP, VOIP,.....