OSI Model – Layers 4-7

INFO-6078 – Managing Enterprise Networks

FANSHAWE

4 – The Transport Layer

7	Application	High-level APIs that provide access to network resources
6	Presentation	Data translation services including encoding, compression and encryption
5	Session	Management of communication sessions
4	**Transport**	**Provides segmentation and process-to-process message delivery**
3	Network	Provides routing and node-to-node delivery
2	Data Link	Error-free transmission of frames between nodes
1	Physical	Transmission of bits over a medium; includes mechanical and electrical specifications

4 – The Transport Layer

- The transport layer is responsible for segmentation, establishing and maintaining temporary communication channels between application processes on different hosts
- Additionally, the transport layer may provide mechanisms for reliable delivery of messages, error-checking and flow control
- In the TCP/IP suite of protocols, two transport layer protocols are available to meet different reliability requirements:
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)

4 – The Transport Layer - Segmentation

- Segmentation allows communication streams from multiple users and applications to share the network by means of multiplexing or interleaving
- Each segment is treated as a separate message and will require addressing and control information for delivery to a destination

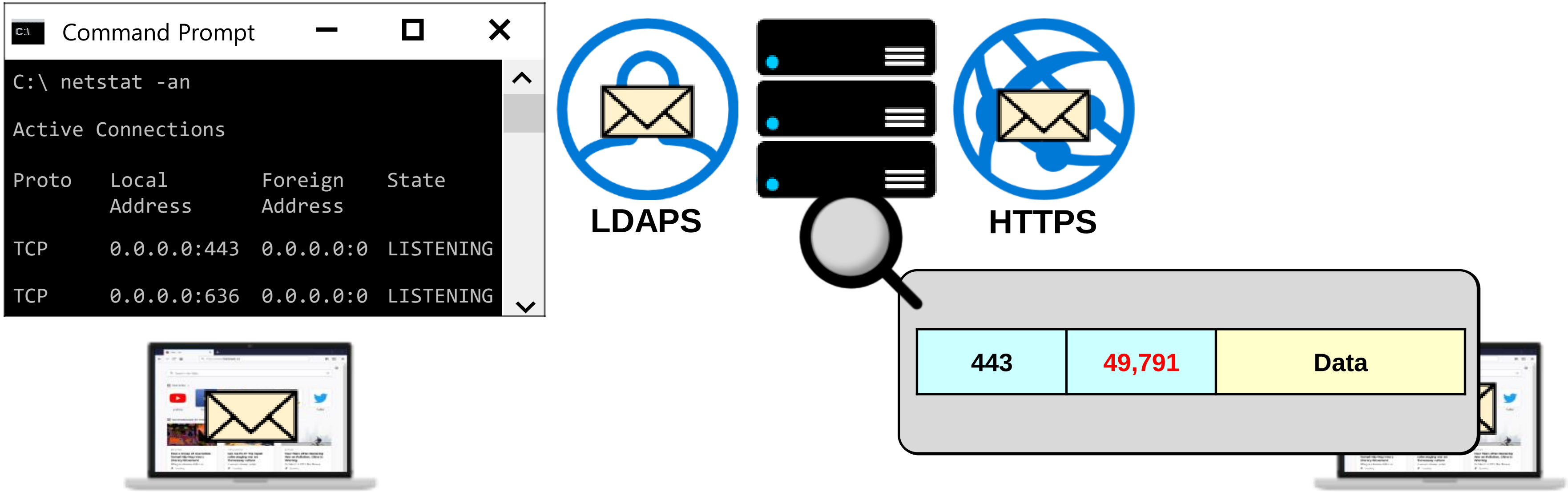
4 – The Transport Layer – Port Numbers

- Both TCP and UDP track connections with the use of source and destination port numbers that range from 0 - 65,535
- The Internet Assigned Numbers Authority (IANA) maintains the registry of assigned port numbers
- Ports are divided into ranges, detailed below:

Port Range	Description
0 to 1,023	Well-Known Ports
1,024 to 49,151	Registered Ports
49,152 to 65,535	Dynamic/Private/Ephemeral Ports

4 – The Transport Layer – Port Numbers

- Services awaiting connections normally use ports in the well-know or registered range
- Client devices select ports from the Dynamic/Private/Ephemeral range to set as the source port



4 – The Transport Layer – Port Numbers

Port #	TCP	UDP	Description
22	✓		Secure Shell (SSH)
53	✓	✓	Domain Name System (DNS)
67/68		✓	Dynamic Host Configuration Protocol (DHCP)
80/443	✓		Hypertext Transfer Protocol (HTTP) / HTTP over TLS/SSL (HTTPS)
80/443		✓	Quick UDP Internet Connections (QUIC)
123		✓	Network Time Protocol (NTP)
161/162	✓	✓	Simple Network Management Protocol (SNMP)
1812/1813	✓	✓	RADIUS authentication/accounting protocol
3389	✓	✓	Microsoft Terminal Server (RDP)
5060	✓	✓	Session Initiation Protocol (SIP)

Transmission Control Protocol (TCP)

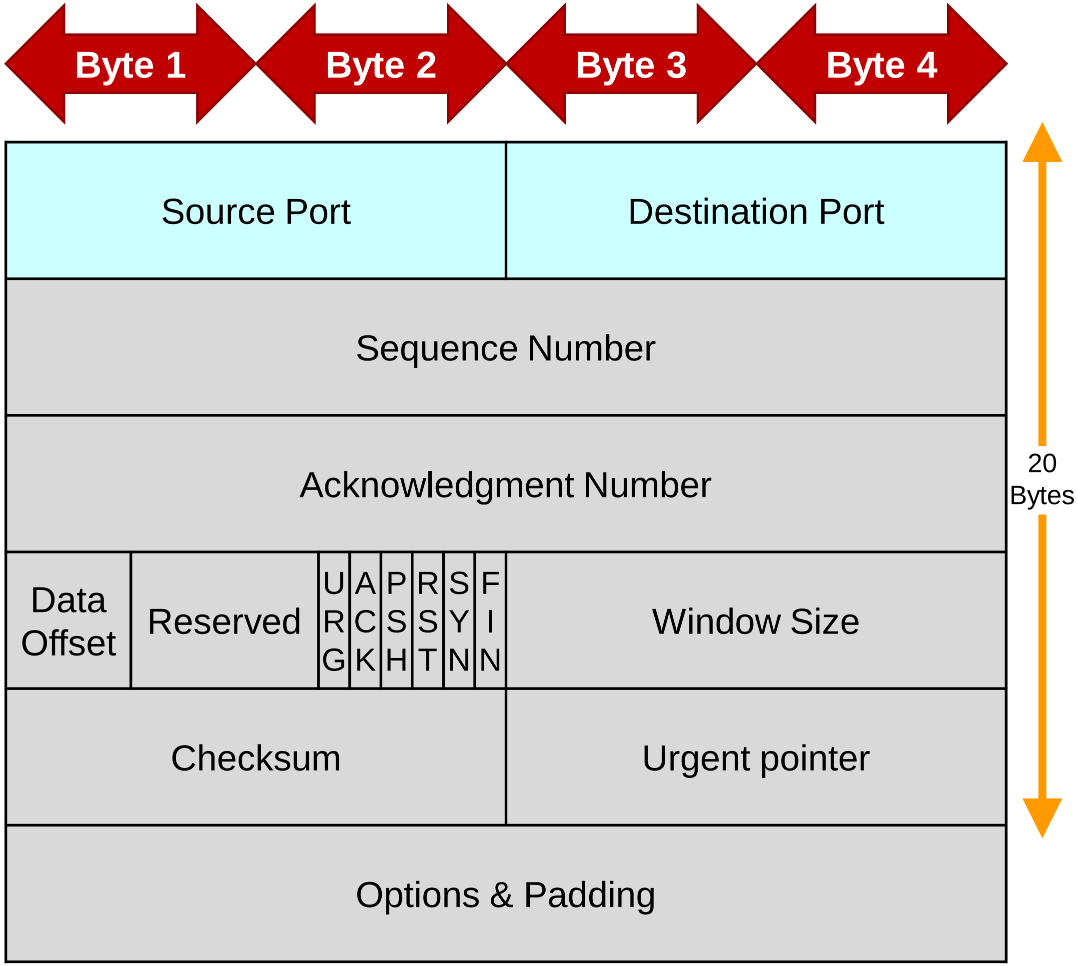
- Provides connection-oriented, stateful delivery of information
- TCP can correct many of the issues related to lower layer delivery of information such as lost, duplicate, or out-of-order packets
- TCP must open a communication session before application data can be transferred
- When the session is no longer needed, TCP must close the connection

Transmission Control Protocol (TCP)

TCP is used as the transport layer protocol when the communication requires one or more of the following:

- Guaranteed data delivery
  - Checksums and timers ensure corrupt or lost segments are retransmitted
- Ordered data delivery
  - Ensures segments are delivered up the OSI model in the same sequence in which they were transmitted
- Session flow control
  - Flow control manages individual communication channels to ensure the receiving device does not get overwhelmed with data

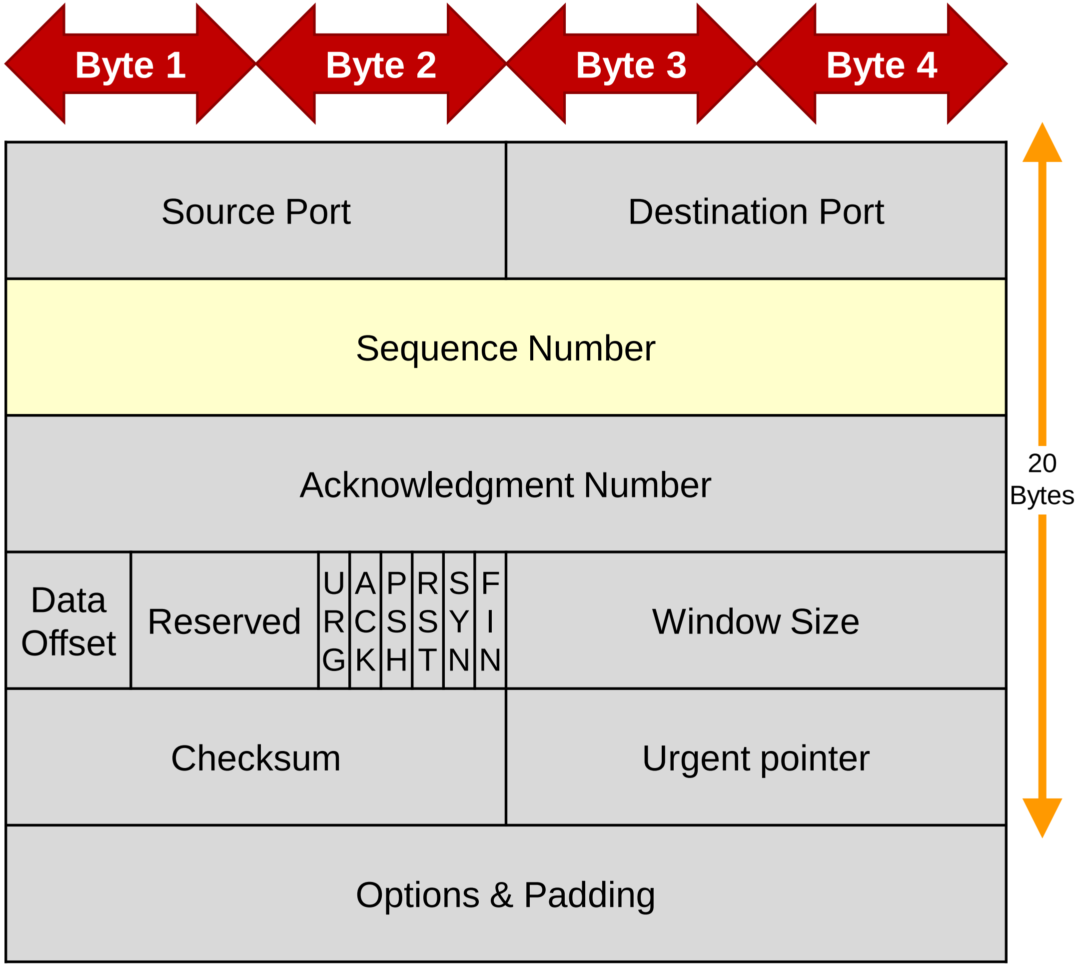
Transport Layer Headers – TCP



Source/Destination Port 16 bits:

- Identifies the sending and receiving processes

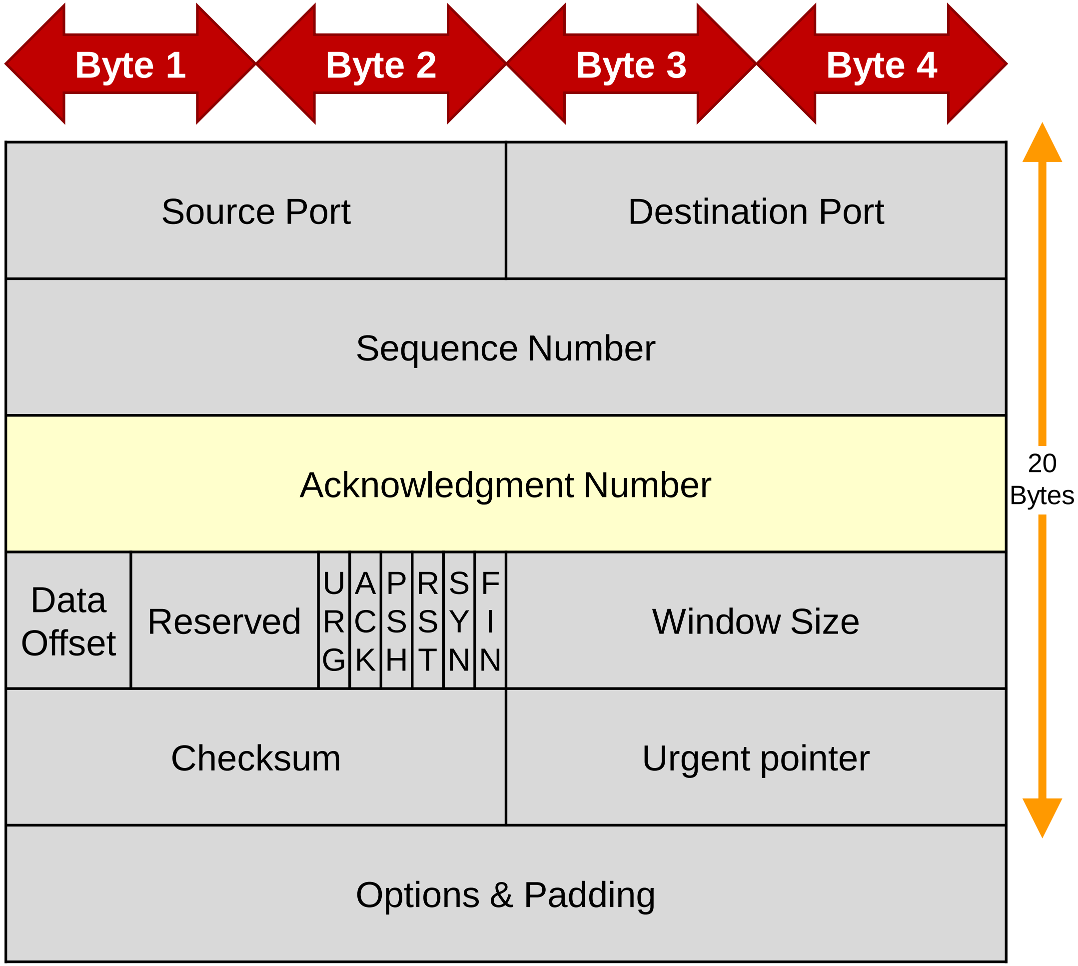
Transport Layer Headers – TCP



Sequence Number
32 bits:

- The sequence number identifies the first byte in the segment
- If the SYN flag is set, the sequence is the Initial Sequence Number (ISN)

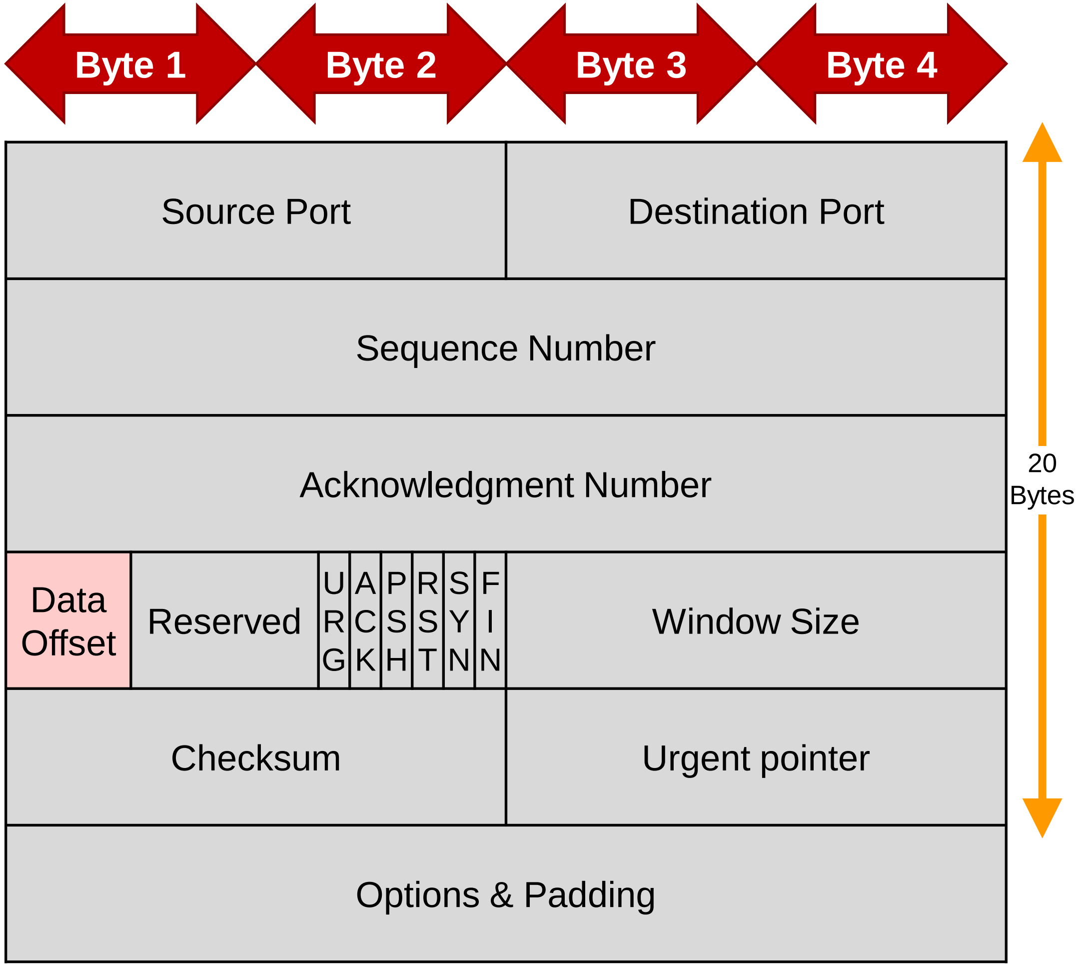
Transport Layer Headers – TCP



Acknowledgment Number
32 bits:

- If the ACK flag is set, the acknowledgement contains the value of the next sequence number the sender of the segment is expecting to receive
- This also acknowledges receipt of all data segments before this value

Transport Layer Headers – TCP



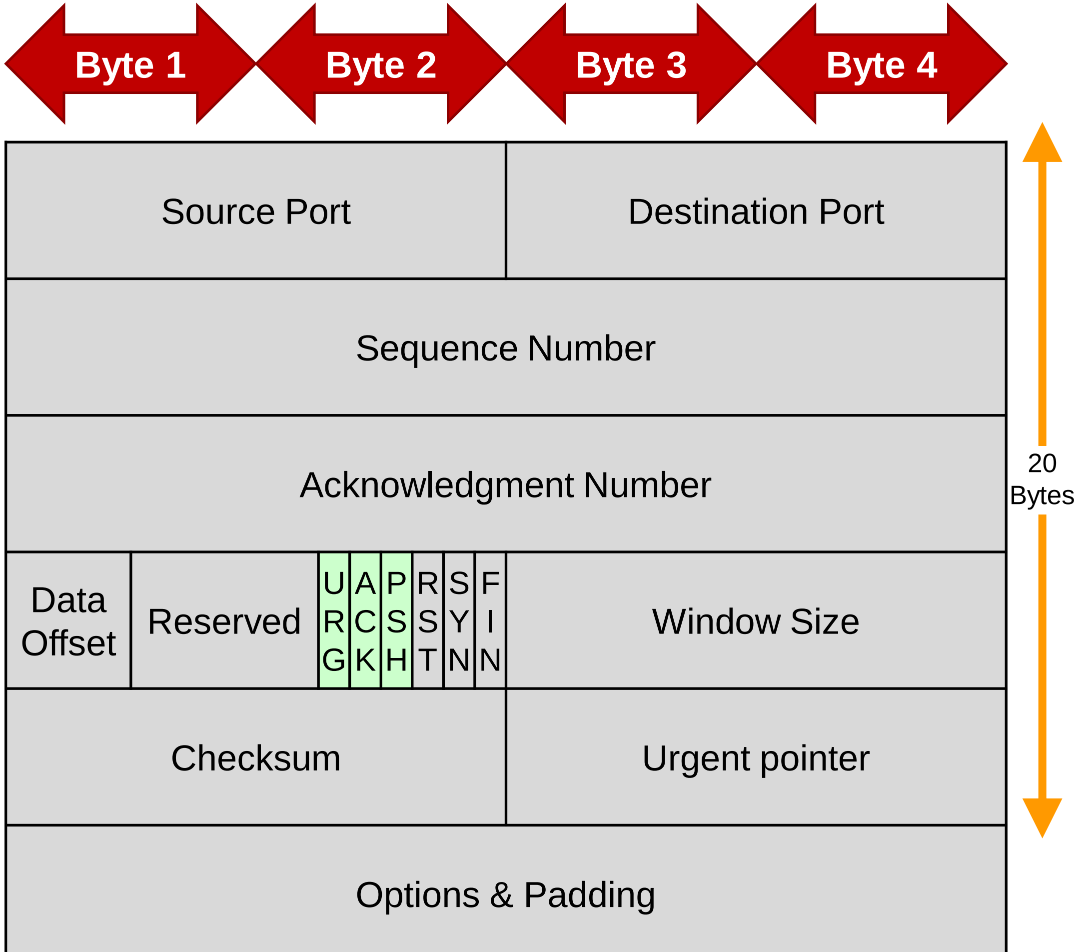
Data Offset 4 bits:

- Sets the number of 32 bit words in the header
- Indicated where the data begins

Reserved 6 bits:

- Reserved for future use
- The latter 3 bits are specified in RFC 3540/RFC 3168 and used for congestion control

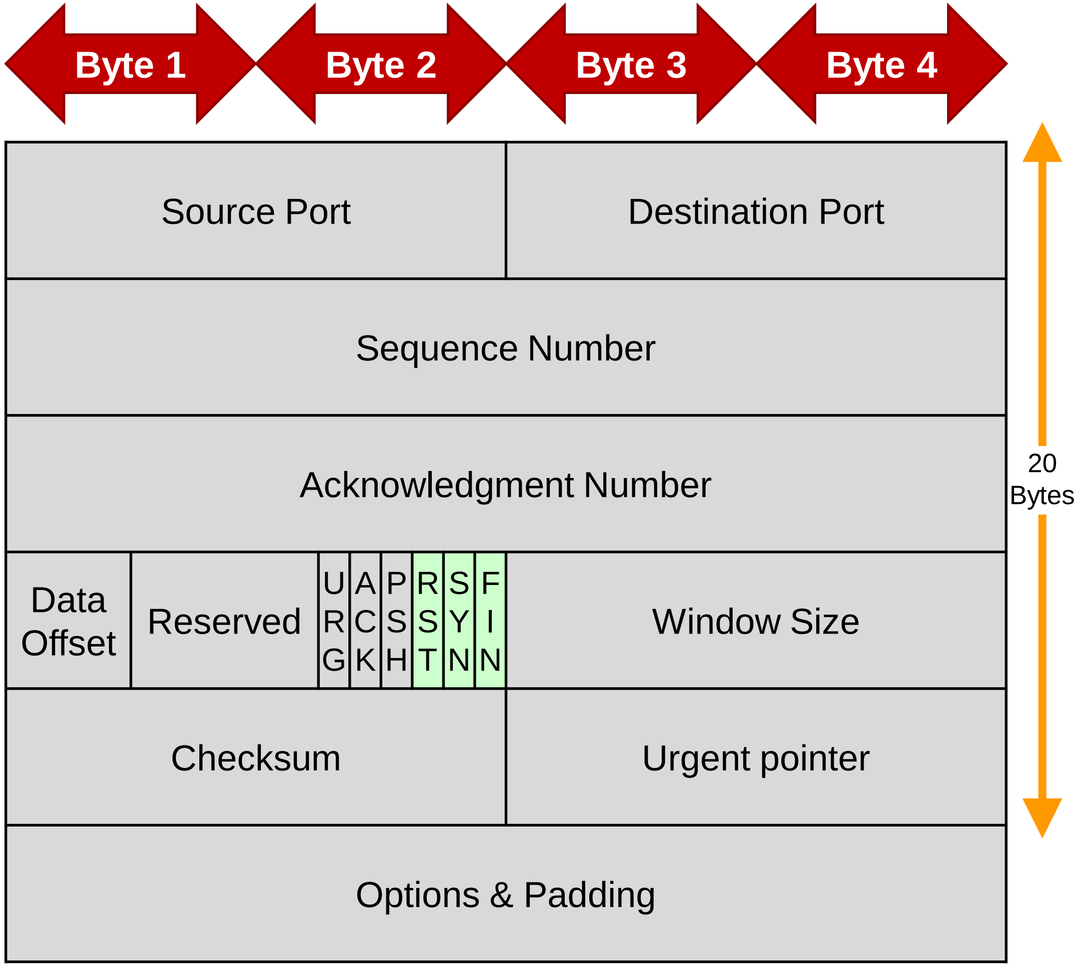
Transport Layer Headers – TCP



Flags 6 bits:

- **URG** – Urgent Pointer field significant
- **ACK** – Acknowledgment field significant
- **PSH** – Push Function: Push buffered data to the application

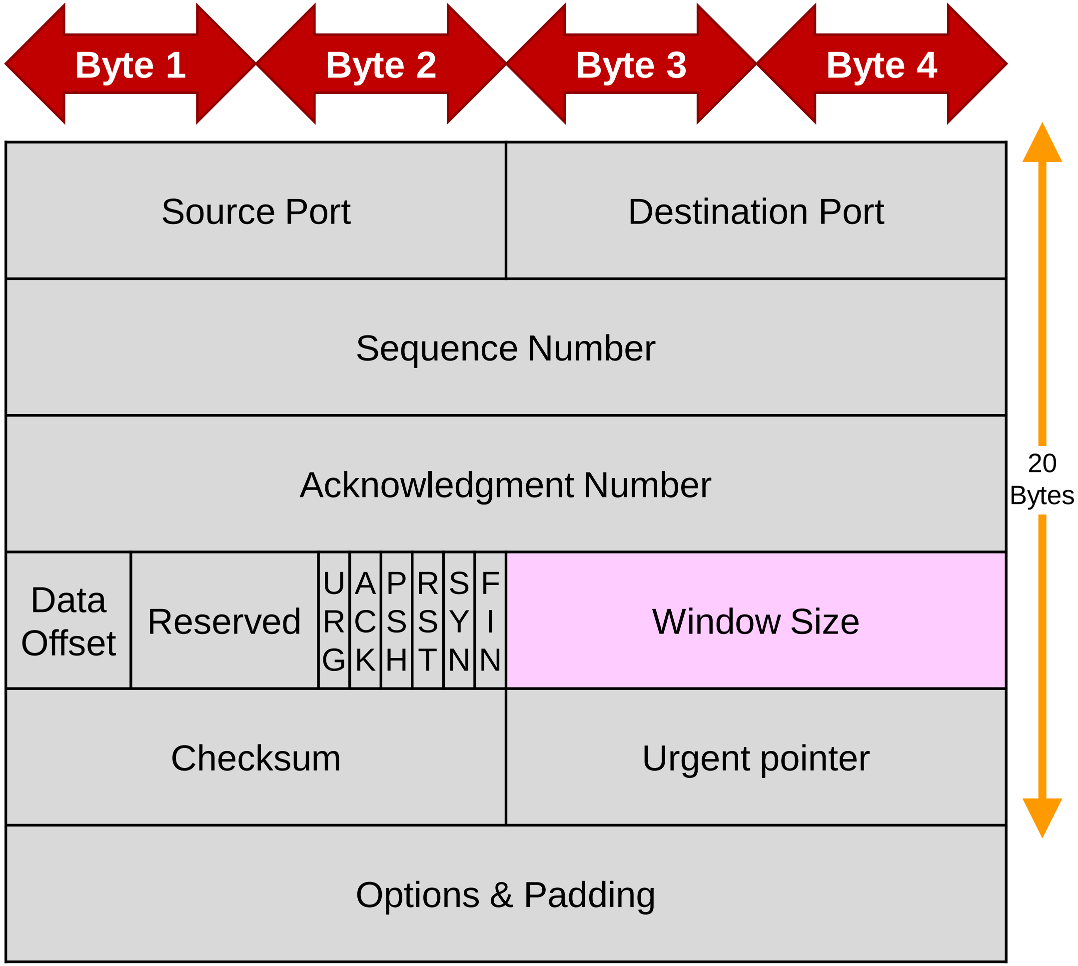
Transport Layer Headers – TCP



Flags 6 bits:

- **RST** – Reset the connection
- **SYN** – Synchronize sequence numbers
- **FIN** – No more data from sender

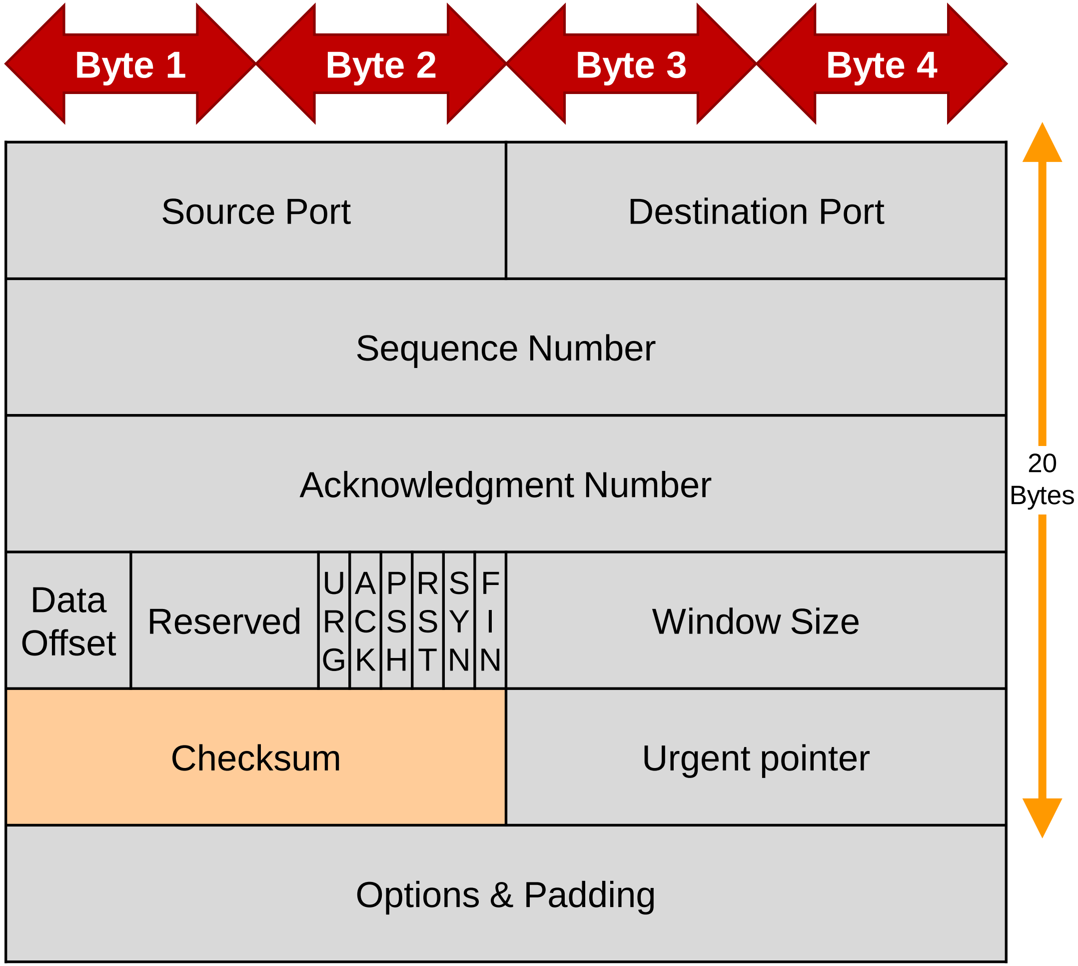
Transport Layer Headers – TCP



Window Size 16 bits:

- The amount of data in bytes beyond that identified in the acknowledgement field that the sender is willing to accept

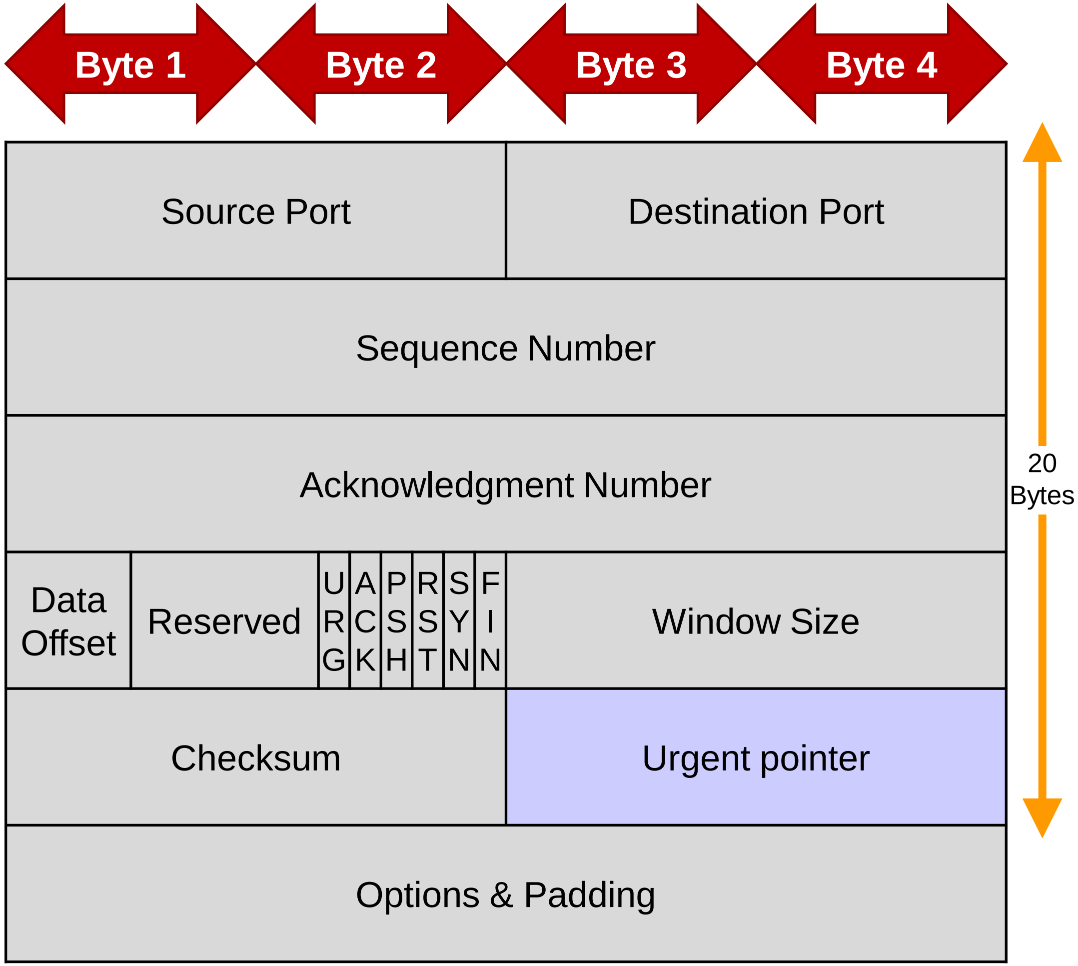
Transport Layer Headers – TCP



Checksum 16 bits:

- Detects corruption and transmission errors based on a pseudo-header
- The pseudo-header contains:
  - Source/Destination IP Addresses
  - Protocol Field
  - Length of the TCP header and payload

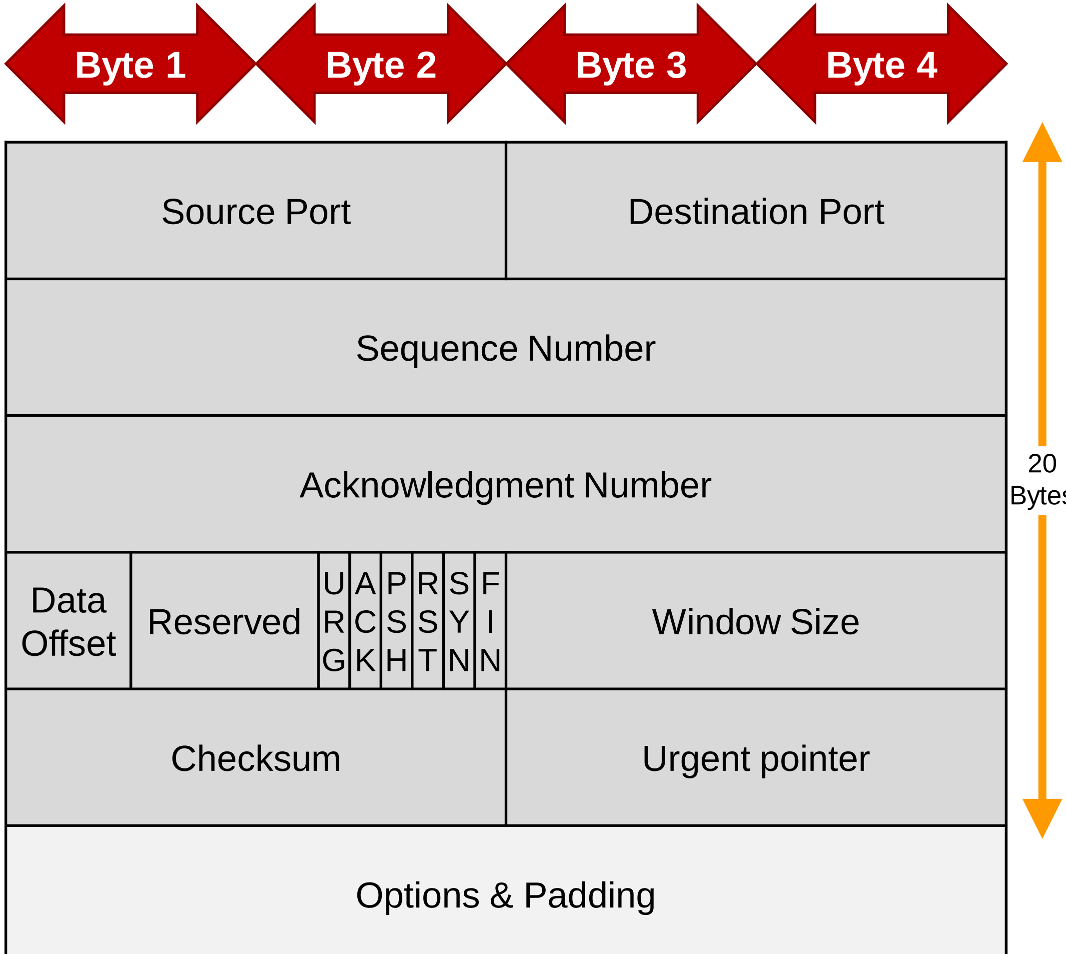
Transport Layer Headers – TCP



Urgent Pointer 16 bits:

- Indicates the end of the urgent data offset in relation to the sequence number
- Only used if the URG flag is set

Transport Layer Headers – TCP



Options varies:

- Used to communicate various control information such as: Maximum Segment Size or Window Scale

Padding varies:

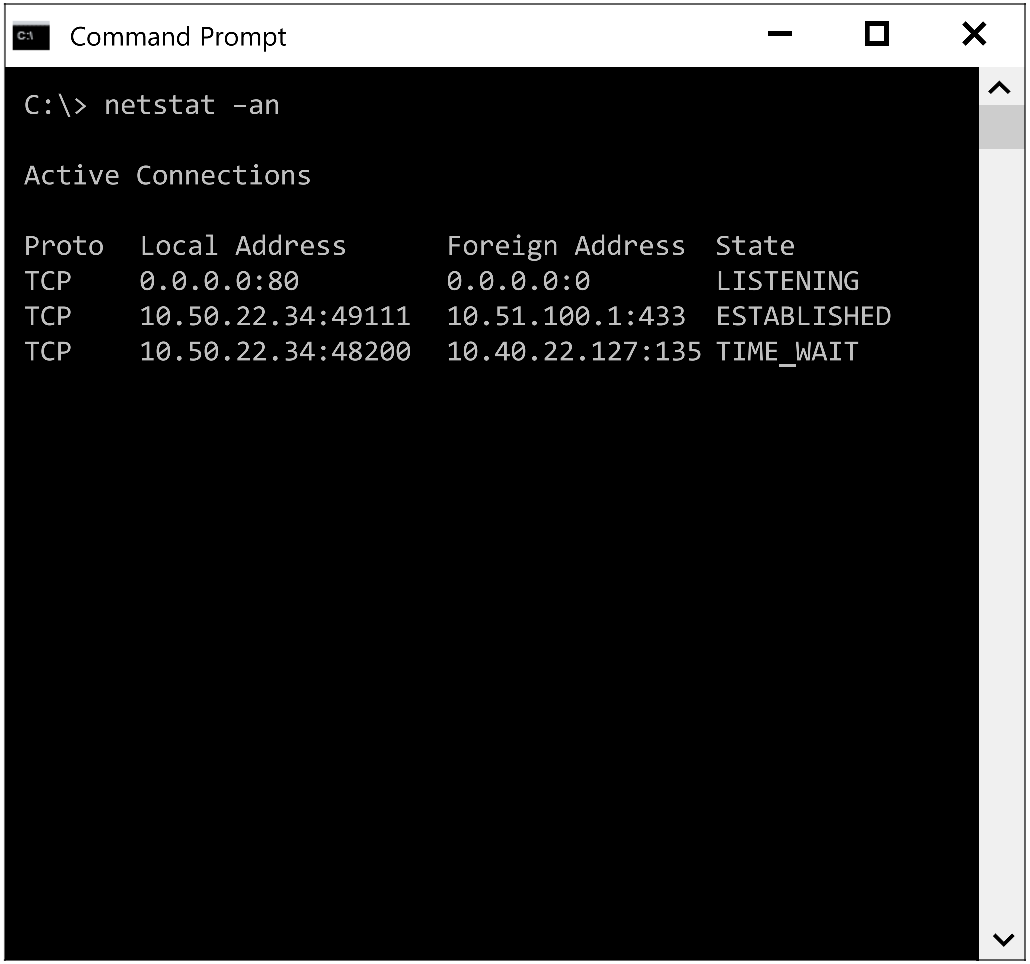
- A number of zeros that is added when options are used to ensure the segment is a multiple of 32 bits

TCP Session and Connection Management

- TCP must setup sessions before application data can be transferred
- Additionally, sessions must be managed and kept in an “alive” state to continue to transfer information
- Session termination occurs when TCP connections are no longer required, or sessions reach a timeout limit

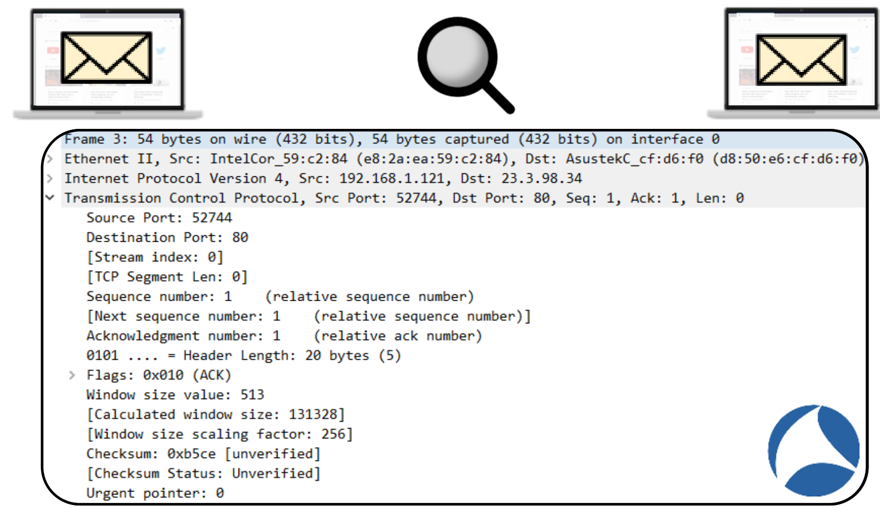
TCP Session Establishment

- TCP uses a process known as the three-way handshake to establish a connection with a remote device
- Before this can occur, a service must bind to a port and listen for incoming connection requests
- Use the flags SYN, SYN/ACK and ACK



TCP Session Establishment

- **Step 1 (SYN)**
  - Host A initiates an active open by sending a SYN request to a server
- **Step 2 (SYN+ACK)**
  - Host B acknowledges Host A’s request by incrementing the Host A’s sequence number
  - Host B also requests a return session by setting the SYN flag
- **Step 3 (ACK)**
  - Host A acknowledges Host B’s request by incrementing Host B’s sequence number

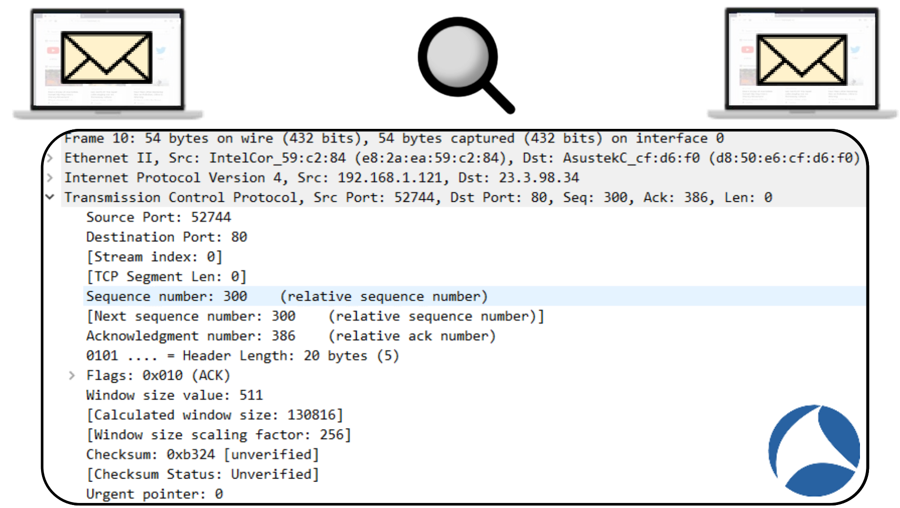


- When a SYN is received, the host starts the **SYN-RECEIVED** timer, allowing 75 seconds for the handshake to complete

TCP Session Termination

- **Step 1 (FIN)**
  - When host A is ready to terminate the session it sends a TCP segment with the FIN flag set to host B
- **Step 2 (FIN+ACK)**
  - Host B acknowledges the request by responding and setting the ACK flag
  - Host B also sends a FIN request to close the connection
- **Step 3 (ACK)**
  - Host A acknowledges Host B's FIN by responding with an ACK

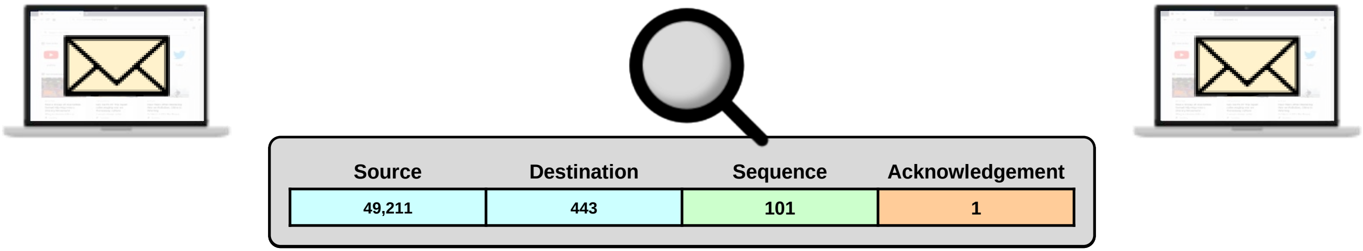
TCP session termination can be in the form of a three-way or four-way handshake



- The host that closed the connection starts the **Time Wait** timer, which allows any outstanding segments to be received.

TCP Sequence Numbers & Acknowledgements

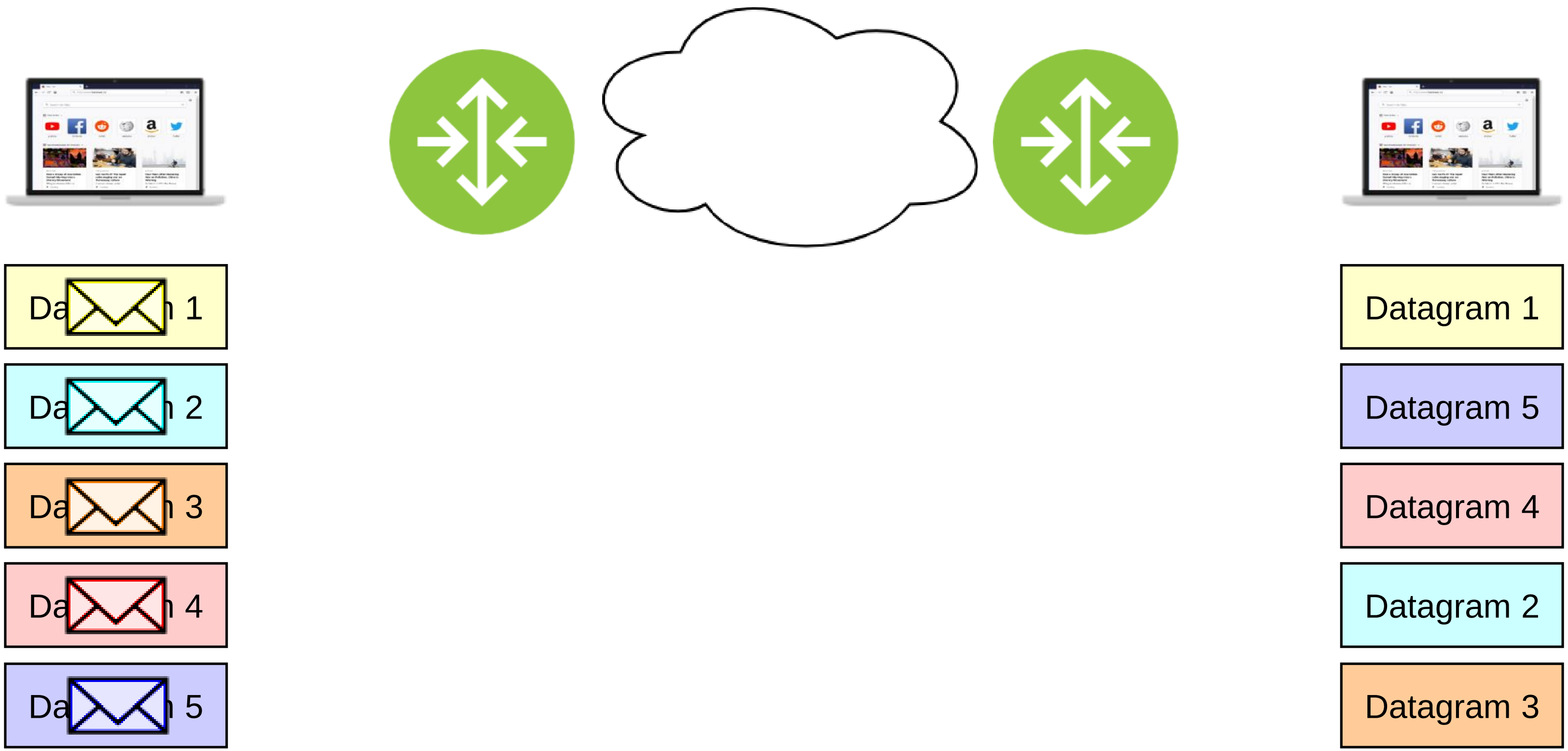
- TCP uses sequence numbers to identify every byte of data
- As each byte of data leaves the source host, the sequence number is incremented to match the amount of data transmitted
- Destination hosts confirm receipt of data using the acknowledgement field in the TCP header
- Sequence numbers are also used to reorder received data should it arrive out of sequence



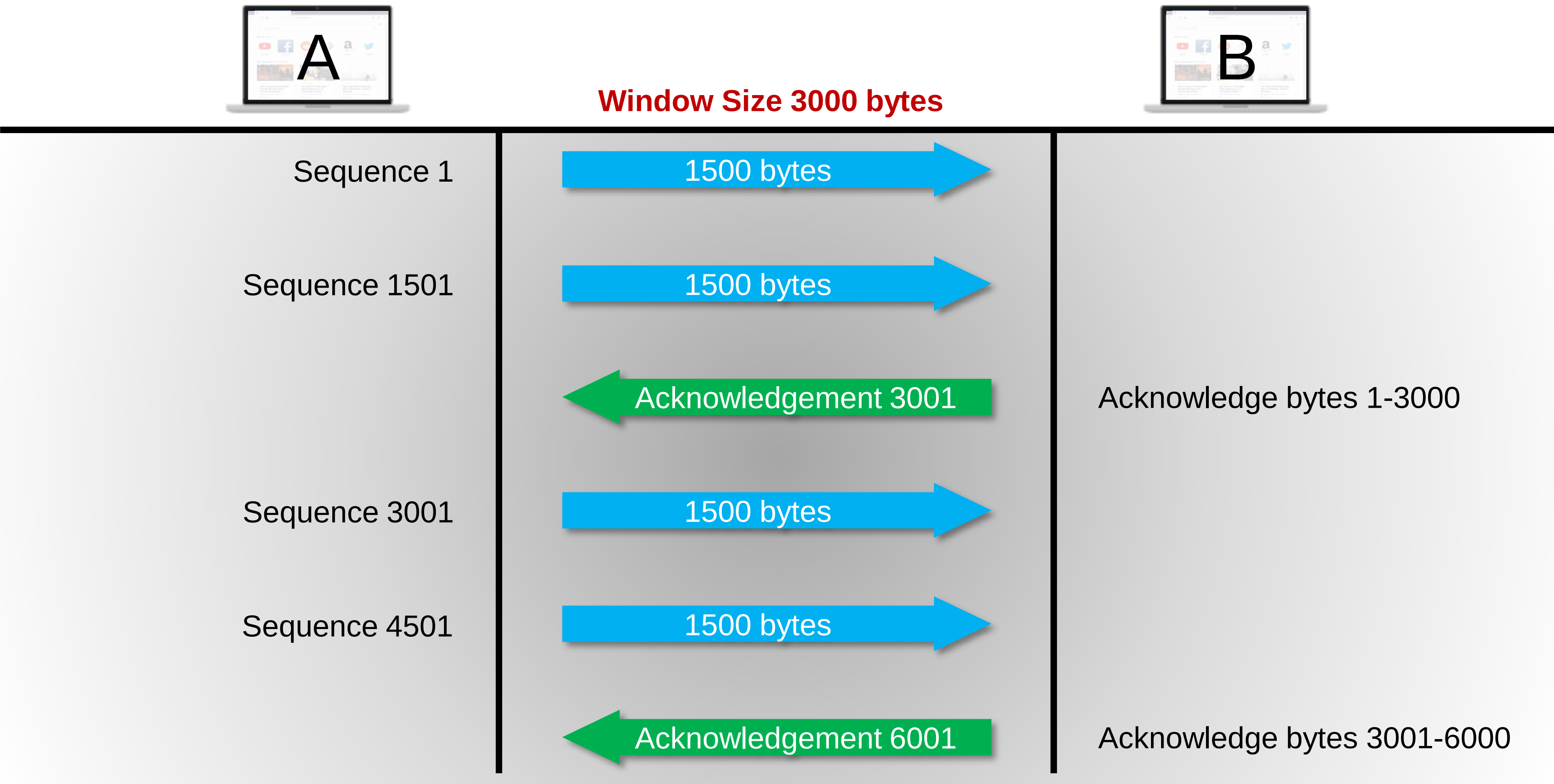
TCP Sequence Numbers & Acknowledgements

- When retransmissions occur, sequence numbers are used to identify duplicate transmissions
- In RFC 793, the initial sequence number is generated based on a clock that increments every 4 microseconds and resets every 4.55 hours (2^{32})
- This provided attackers the ability to guess sequence numbers based and have the ability to hijack TCP sessions
- Newer implementations of TCP use a random number generator or other methods to choose the ISN

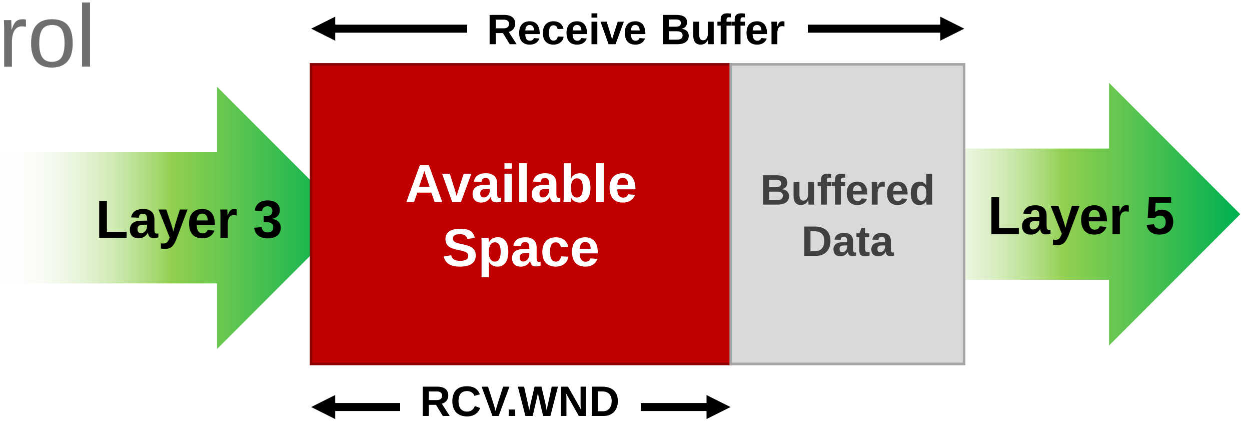
TCP Sequence Numbers & Ordered Delivery



TCP Window Size

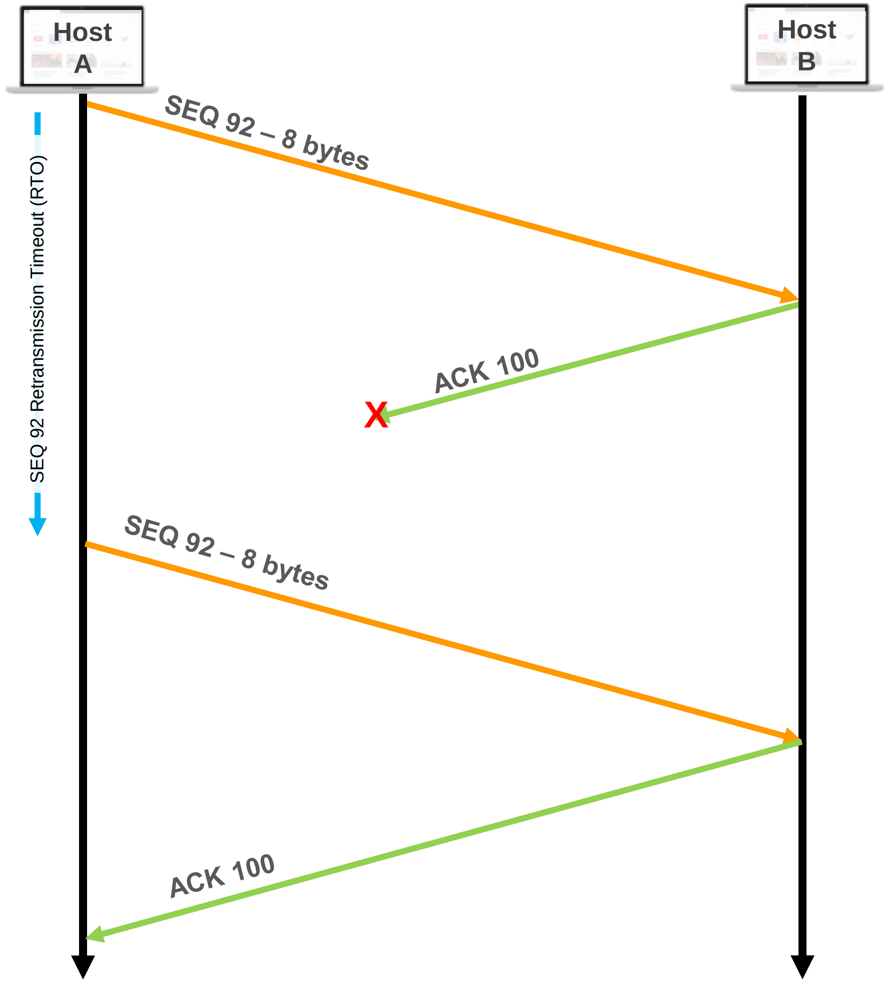


TCP Flow Control

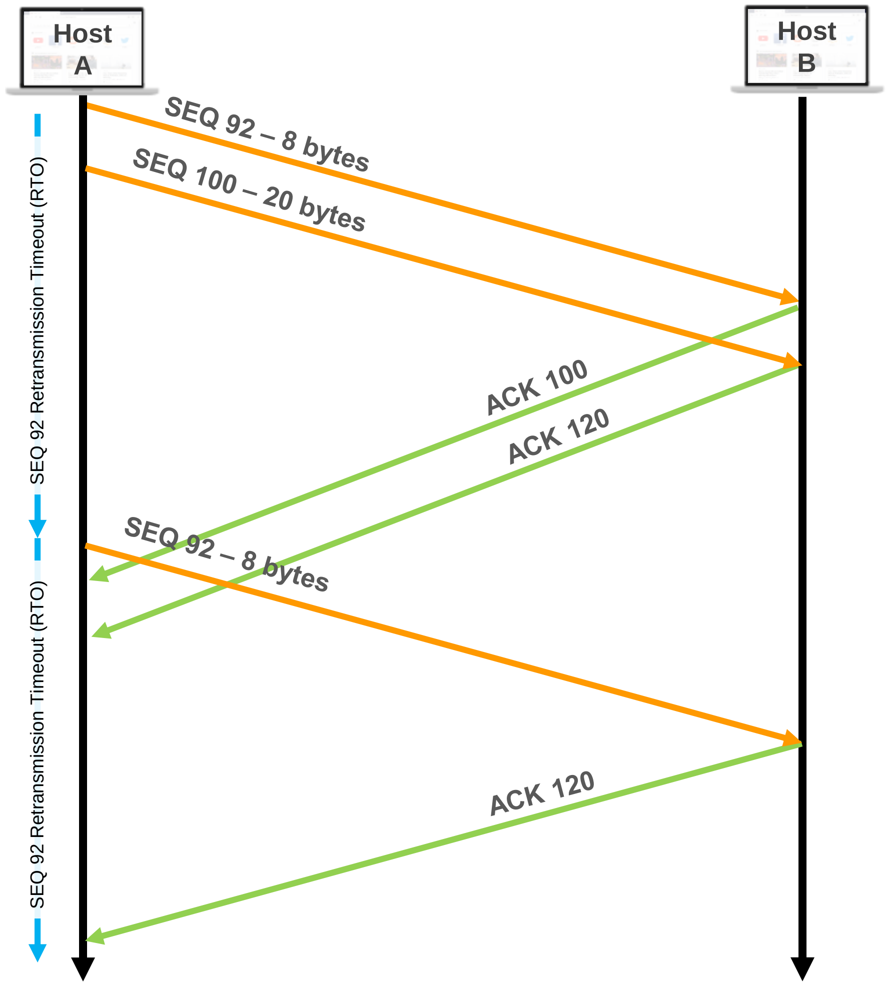


- TCP utilizes end-to-end flow control mechanisms to limit the amount of data a sender can transmit; to prevent it from overwhelming the receiving device
- To accomplish this, the receiving device advertises the amount of available space in it’s buffer to the sending device in a value known as the receive window
- When the receiver advertises a receive window of 0, the sending device stops sending data and starts the persist timer, which is employed to prevent a deadlock situation

TCP Retransmission – Lost ACK



TCP Retransmission – Premature Timeout

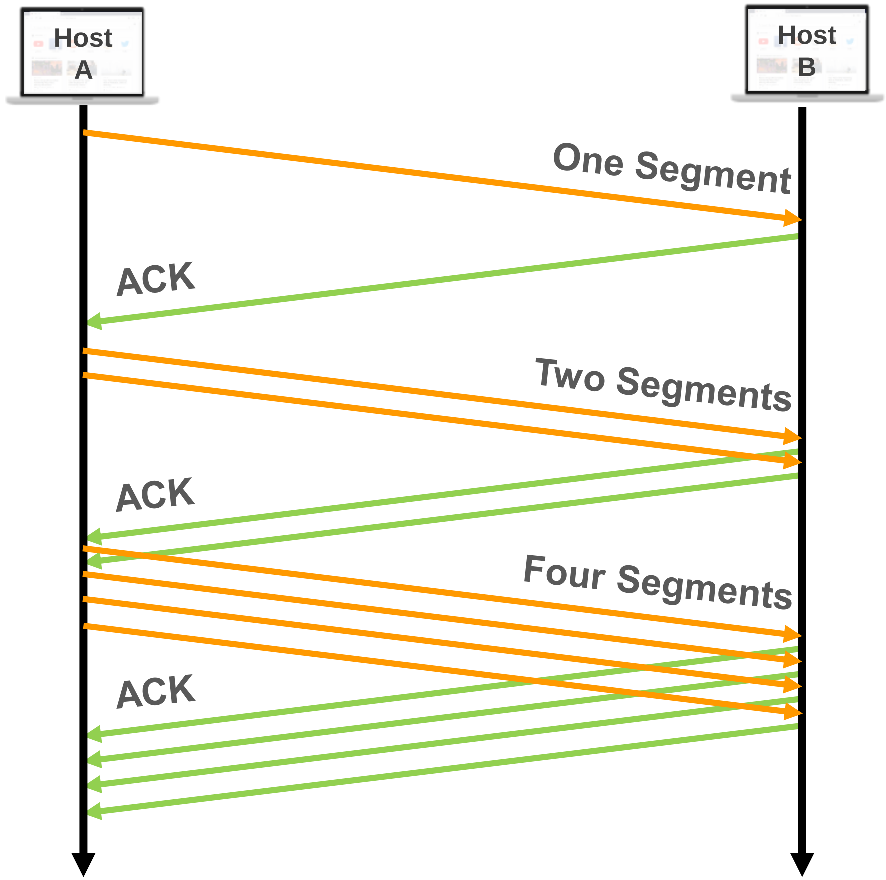


TCP Congestion Control

- Updated with RFC5681, TCP utilizes four mechanisms to improve performance and provide congestion control
- The congestion detection in TCP is decentralized, and comes from inferred feedback from the TCP hosts:
  - if transmitted segments are acknowledged, there is no congestion
  - If transmitted segments are not acknowledged, the network is congested
- The algorithms that provide congestion control are:
  - Slow-Start
  - Congestion Avoidance
  - Fast Retransmit
  - Fast Recovery

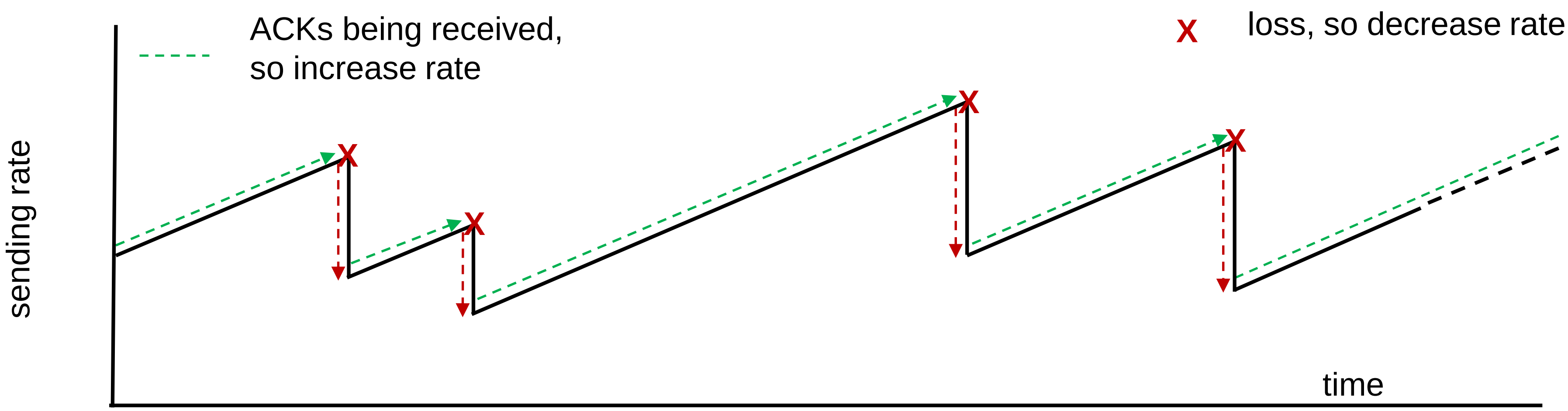
TCP Slow-Start

- When the connection commences, slow start has a congestion window of 1, 2, 4 or 10 maximum segment size (MSS)
- Each time transmitted segments are acknowledged, the congestion window is adjusted by that amount, essentially doubling the amount of transmitted data until a loss is detected, or the limit of the Receive Window is reached



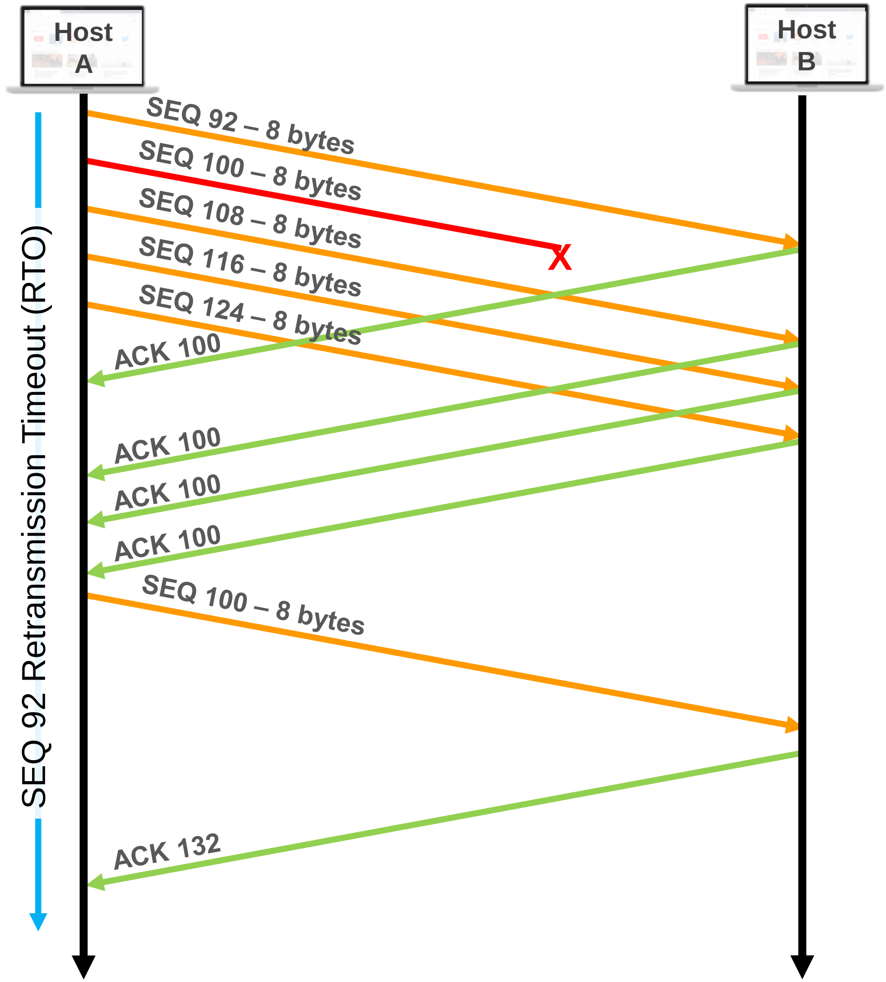
TCP Congestion Avoidance

- Congestion avoidance is closely tied to Slow-Start.
- Slow-Start is used to increase the size of the congestion window, but congestion avoidance manages the window size
- When a loss is detected, the congestion window is cut in half and slow-start is restarted to increase the window size



TCP Fast Retransmit & Fast Recovery

- Fast Retransmit allows a sending host to reduce the amount to time it takes to recover from a lost segment
- If three or more duplicate acknowledgements are received, fast retransmit sends the next segment again
- Fast Recovery is invoked until all transmitted segments are acknowledged



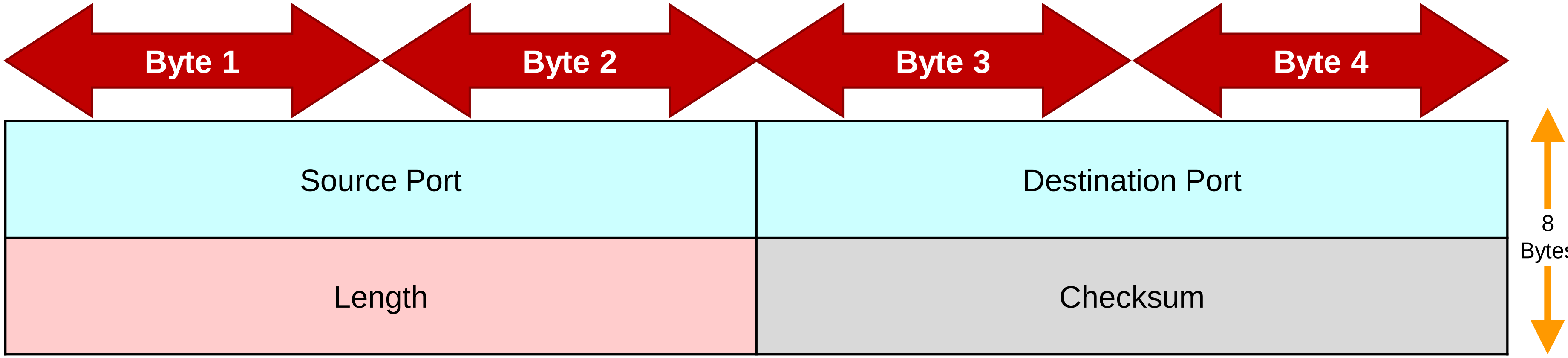
User Datagram Protocol (UDP)

- Originally defined in RFC 768
- UDP is a connectionless protocol and no session is established before application data is sent

UDP is used as the transport layer protocol when the communication requires one or more of the following:

- Fastest Possible Delivery
  - UDP is ideally suited to communications that are time-sensitive
- Tolerance to Dropped/Lost Datagrams
  - UDP does not provide mechanisms to retransmit drop or lost datagrams
- Tolerance to Unordered Datagrams
  - UDP does not reorder datagram delivery before passing received data up the OSI model

Transport Layer Headers – UDP



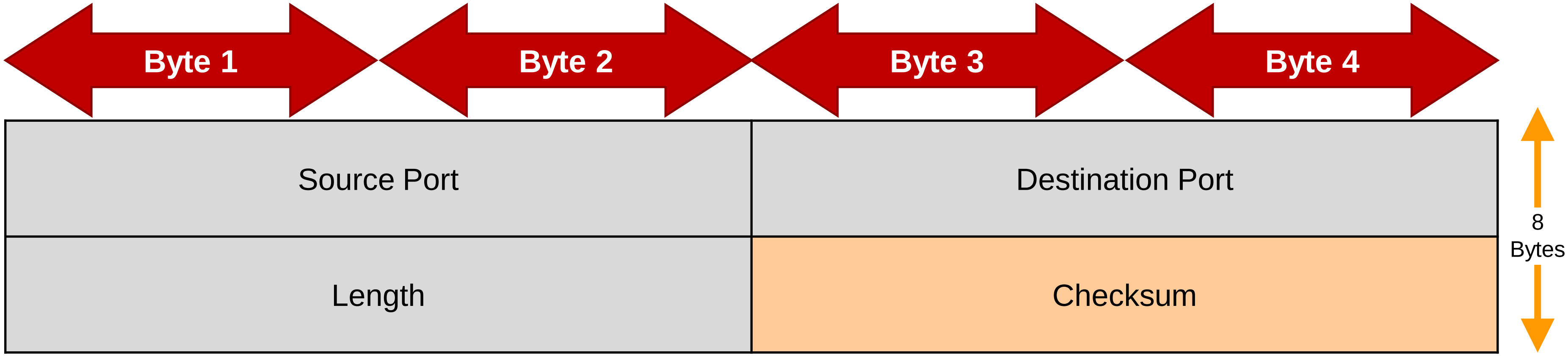
Source/Destination Port 16 bits:

- Identifies the sending and receiving processes

Length 16 bits:

- The size of the header and data in bytes

Transport Layer Headers – UDP



Checksum 16 bits:

- Detects corruption and transmission errors based on a pseudo-header
- The pseudo-header contains:
  - Source/Destination IP Addresses
  - Protocol Field
  - Length of the UDP header and payload

5 – The Session Layer

7	Application	High-level APIs that provide access to network resources
6	Presentation	Data translation services including encoding, compression and encryption
5	Session	Management of communication sessions
4	Transport	Provides segmentation and process-to-process message delivery
3	Network	Provides routing and node-to-node delivery
2	Data Link	Error-free transmission of frames between nodes
1	Physical	Transmission of bits over a medium; includes mechanical and electrical specifications

5 – The Session Layer

- The session layer is responsible for the setup, maintenance and teardown of connections (sessions) between source and destination applications
- The transfer of data is managed by lower layers of the OSI model, but the session layer manages the synchronization of connections
- Transmission modes in the session layer include simplex, half-duplex and full-duplex

6 – The Presentation Layer

7	Application	High-level APIs that provide access to network resources
6	**Presentation**	**Data translation services including encoding, compression and encryption**
5	Session	Management of communication sessions
4	Transport	Provides segmentation and process-to-process message delivery
3	Network	Provides routing and node-to-node delivery
2	Data Link	Error-free transmission of frames between nodes
1	Physical	Transmission of bits over a medium; includes mechanical and electrical specifications

6 – The Presentation Layer

- The presentation layer is responsible for transferring data to endpoints in a format that the endpoint is capable of receiving such as ASCII or binary data streams
- Encryption and compression are commonly performed in the presentation layer
- The presentation layer is also capable of serializing and deserializing of complex data objects

7 – The Application Layer

7	**Application**	**High-level APIs that provide access to network resources**
6	Presentation	Data translation services including encoding, compression and encryption
5	Session	Management of communication sessions
4	Transport	Provides segmentation and process-to-process message delivery
3	Network	Provides routing and node-to-node delivery
2	Data Link	Error-free transmission of frames between nodes
1	Physical	Transmission of bits over a medium; includes mechanical and electrical specifications

7 – The Application Layer

- The application layer enable user and system applications to interface with the network stack and communicate across a network
- Functions of the application layer include identification of communication endpoints, availability determination and interface management
- The OSI model does not include specifications for specific socket interfaces, and their design is implementation specific

References

- UTP Cabling Image - Retrieved from: https://www.flickr.com/photos/33399192@N02/3113089409
- Spectrum Analyzer Image (cropped) – Retrieved from: https://en.wikipedia.org/wiki/File:Bluetooth_signal_behind_wireless_lan_signal.png
- SC Optical Fiber Image – Retrieved from: https://commons.wikimedia.org/wiki/File:SC-optical-fiber-connector-hdr-0a.jpg
- Spectrum Analyzer 5Ghz (modified) – Retrieved from: https://commons.wikimedia.org/wiki/File:SpectrumAnalyzerDisplay.png

References

- Speedtest.net Images - Retrieved from: https://www.speedtest.net
- IP Version 4 – Special Use Address Ranges:
  - https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml
- Session Layer
  - https://searchnetworking.techtarget.com/definition/Session-layer

FANSHAWE