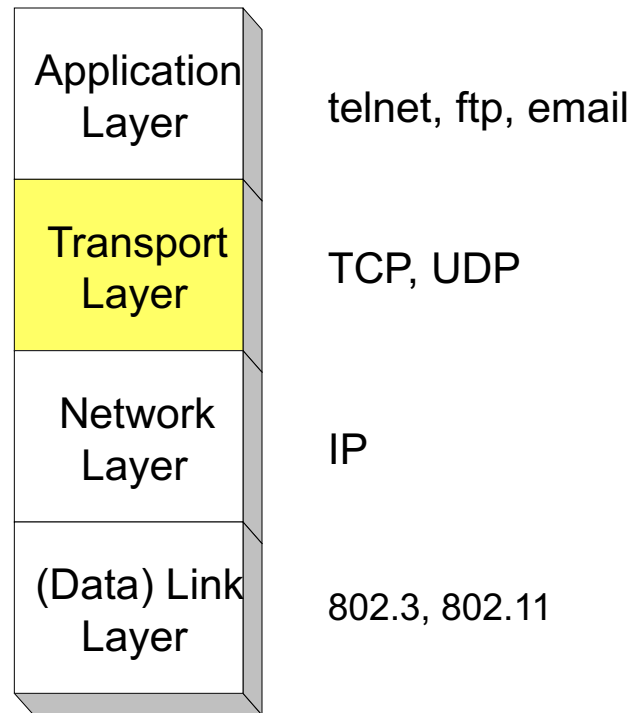


# Introduction to transport layer

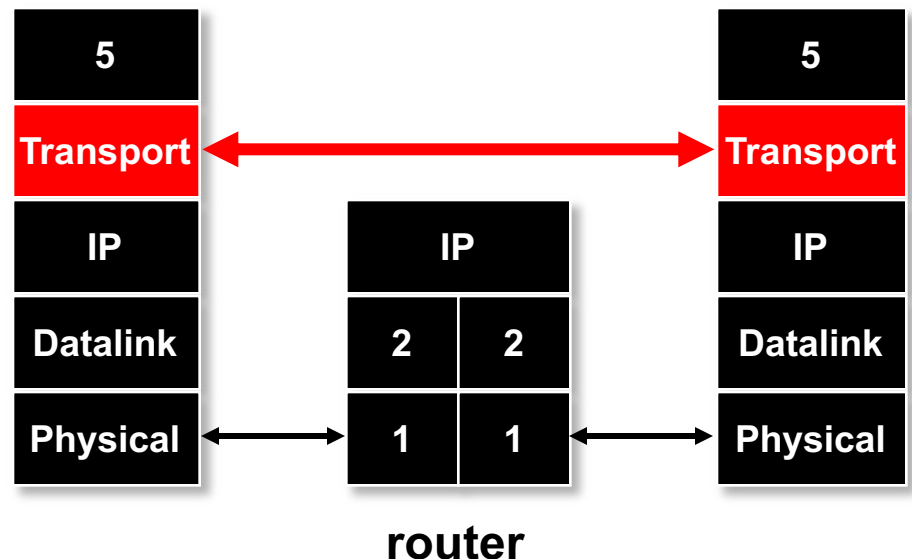
ECE 50863 – Computer Network Systems

# Transport Layer

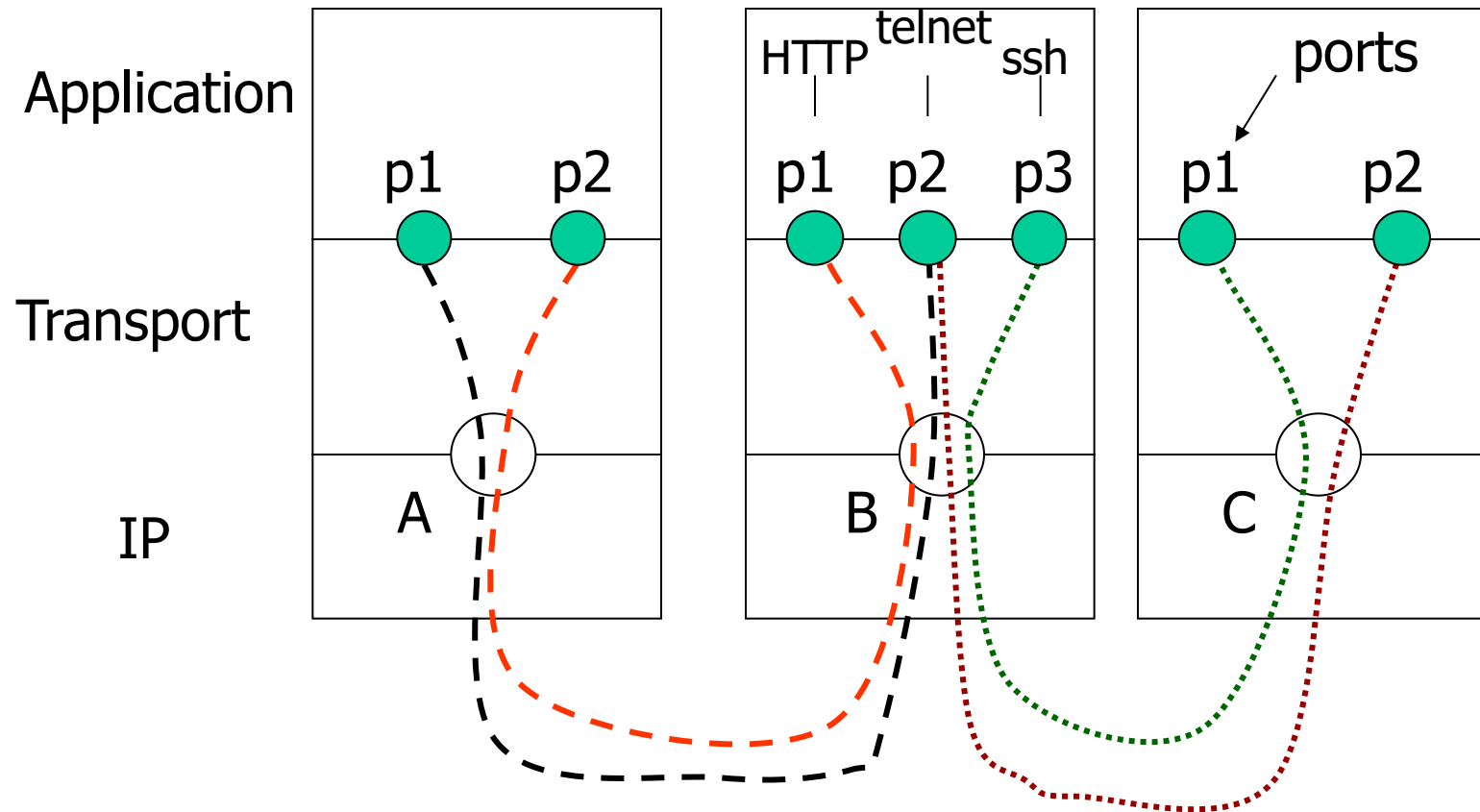


# Transport Protocols Concern only End Hosts, not Routers

- Header generated by sender is interpreted only by the destination
- Routers view transport header as part of the payload



# Using Transport Layer Port Number to (De)multiplex traffic



A TCP/UDP packet contains the following information:  
(Source Address, Destination Address, Source Port, Destination Port)

# Popular Transport Protocols

- UDP:
  - Barebones,minimal
  - Does not provide much functionality besides multiplexing
- TCP:
  - Elaborate, lots of additional functionality provided.

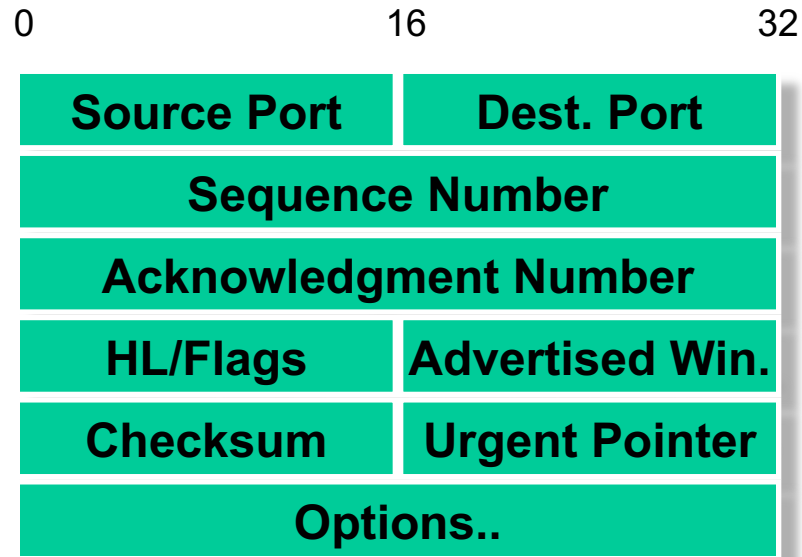
# Using UDP

- Main reasons for using UDP
  - Lower latency
  - Reliable transmission not needed for some applications
  - TCP heavy-weight: application may only need some features
- Examples:
  - Video conferencing (e.g., Zoom)
  - Remote procedure calls
  - Gaming applications.
- Custom protocols/applications can be implemented on top of UDP
  - implement own reliability, flow control, ordering, congestion control
  - Selectively enable only necessary features

# Transmission Control Protocol (TCP)

- Reliable bidirectional in-order byte stream
- Lots of functionality
- Connection establishment.
  - Logical end-to-end connection, connection state to optimize performance
- Error control
  - Hide unreliability of the network layer from applications
  - Many types of errors: corruption, loss, duplication, reordering.
- End-to-end flow control and congestion control
  - Avoid flooding the receiver and network.

# TCP Header



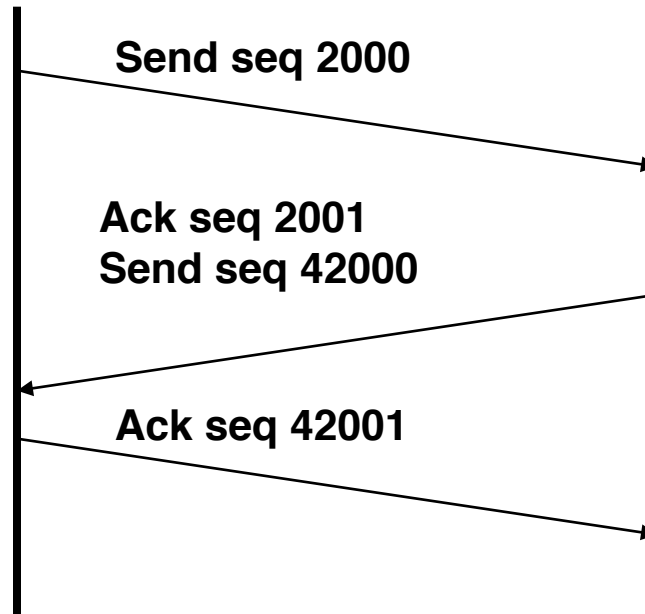
- 20 bytes total
- Sequence Number, ACK: relate to TCP functionality for achieving reliable delivery.
- Advertised window: For achieving a function called “flow control”.
- HL: Specifies Header Length
- Flags: 6 flags in all
- Urgent pointer: not common TCP usage, used to signal certain data is “out-of-band” and must be processed immediately



# Important TCP Flags

- SYN: Synchronize
  - Used when setting up connection
- FIN: Finish
  - Used when tearing down connection
- ACK
  - Acknowledging received data
- RESET:
  - Receiver wants to abort connection, as it received unexpected segment.
- Push and Urgent flags:
  - Not as commonly used.
  - Signify receiving process must be notified, or out-of-band data

# Bidirectional Communication

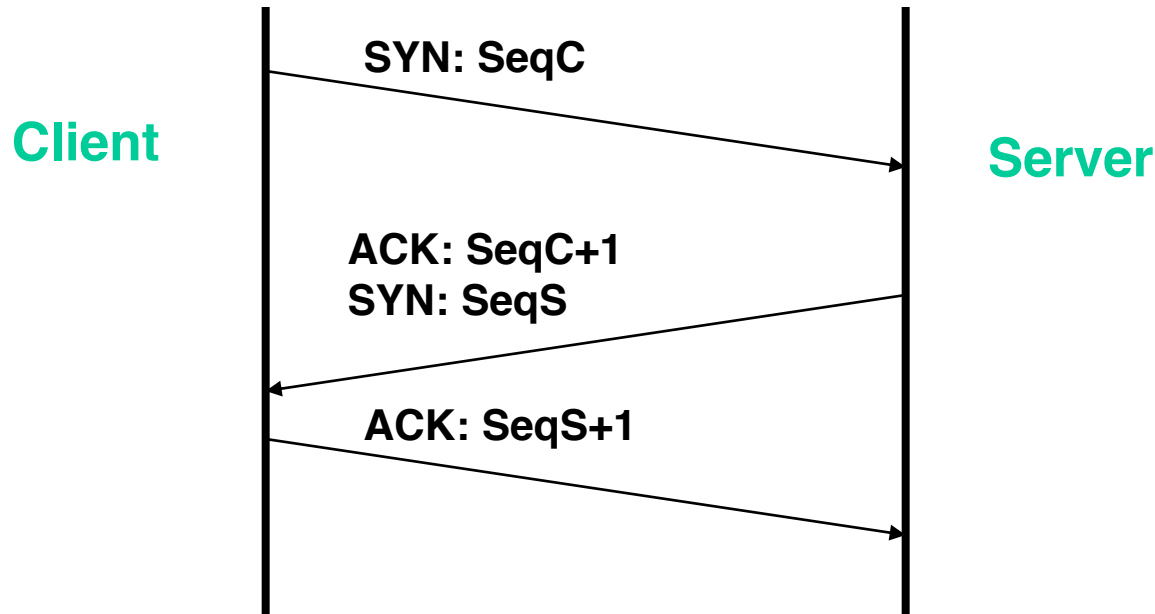


- Each Side of Connection can Send *and* Receive
- What this Means
  - Maintain different sequence numbers for each direction
  - Single segment can contain new data for one direction, plus acknowledgement for other
    - But some contain only data & others only acknowledgement

# Connection Setup

- Why need connection setup?
- Mainly to agree on starting sequence numbers
  - Starting sequence number is randomly chosen
  - Reason, to reduce the chance that sequence numbers of old and new connections from overlapping

# Establishing Connection



- Three-Way Handshake
  - Each side notifies other of starting sequence number it will use for sending
  - Each side acknowledges other's sequence number
    - SYN-ACK: Acknowledge sequence number + 1
  - Can combine second SYN with first ACK

# Tearing Down Connection

- Either Side Can Initiate Tear Down
  - Send FIN signal
  - “I’m not going to send any more data”
- Other Side Can Continue Sending Data
  - Half open connection
  - Must continue to acknowledge
- Acknowledging FIN
  - Acknowledge last sequence number + 1

