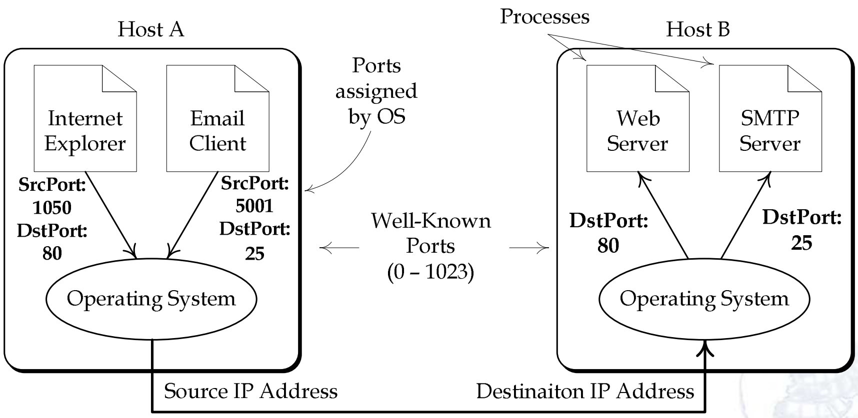
Lecture 18: Transport Layer Protocols (Layer 4)

Dr. Mohammed Hawa
Electrical Engineering Department
University of Jordan

Transport Layer Functions

- **Demultiplexing:** allow multiple application programs on any given host to simultaneously communicate using one host IP address.
- **Divide** large messages into smaller finite-size packets.
- **Reliable end-to-end transmission:** Retransmit dropped packets, re-arrange out-of-order packets, and neglect duplicate packets.
- **Flow control:** prevent the sender from overrunning the capacity of the receiver.
- **Congestion control:** throttle senders to avoid too much data from being injected into the network, thereby causing switches or links to become overloaded.

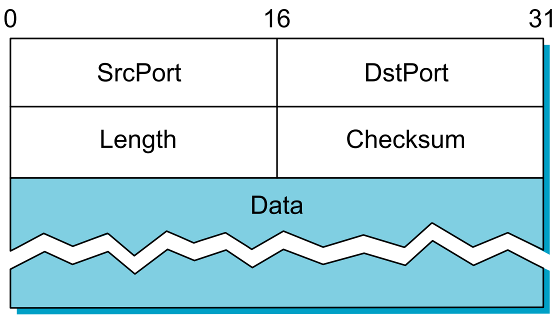
Multiplexing: TCP/UDP Ports



User Datagram Protocol (UDP)

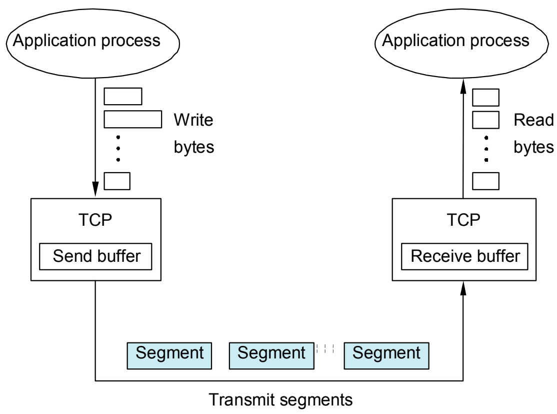
- The simplest possible transport protocol that runs on top of IP.
- Allows applications to send IP datagrams without having to establish a connection.
- Described in RFC 768.
- UDP adds a level of process demultiplexing at the source and destination hosts through port numbers.
- Aside from this, UDP adds no other functionality to the best-effort service provided by the underlying IP network.

UDP Segment Format

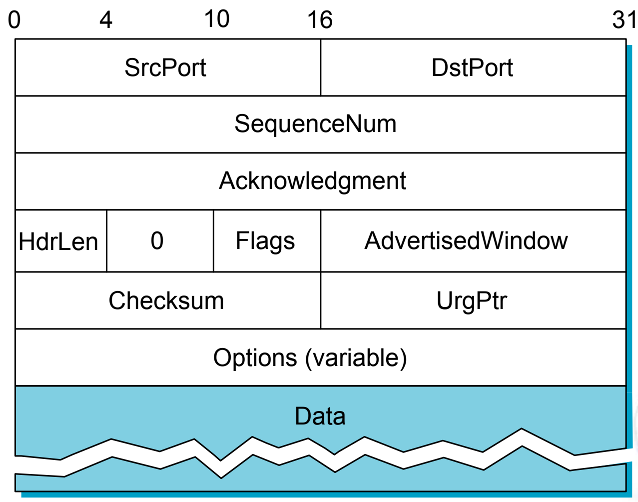


Transmission Control Protocol (TCP)

- TCP provides a reliable, full-duplex, point-to-point connection-oriented byte stream transport service on top of IP.



TCP Segment Format

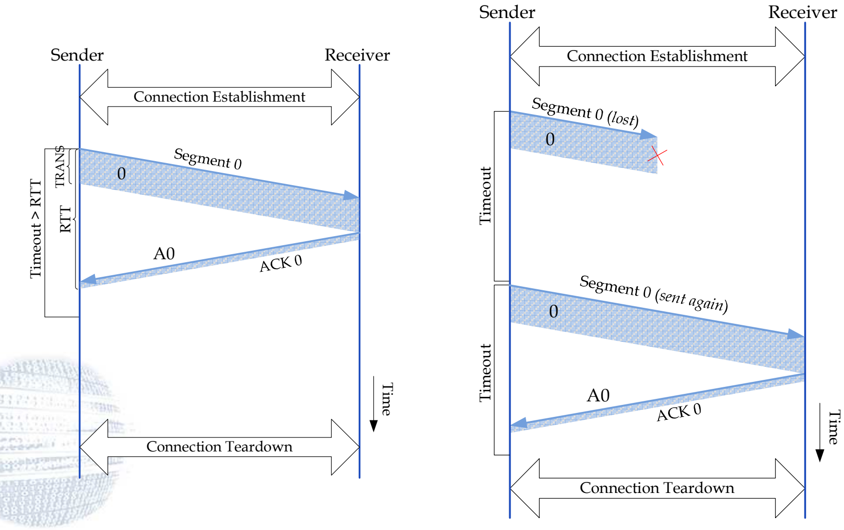


Reliable Delivery of Segments

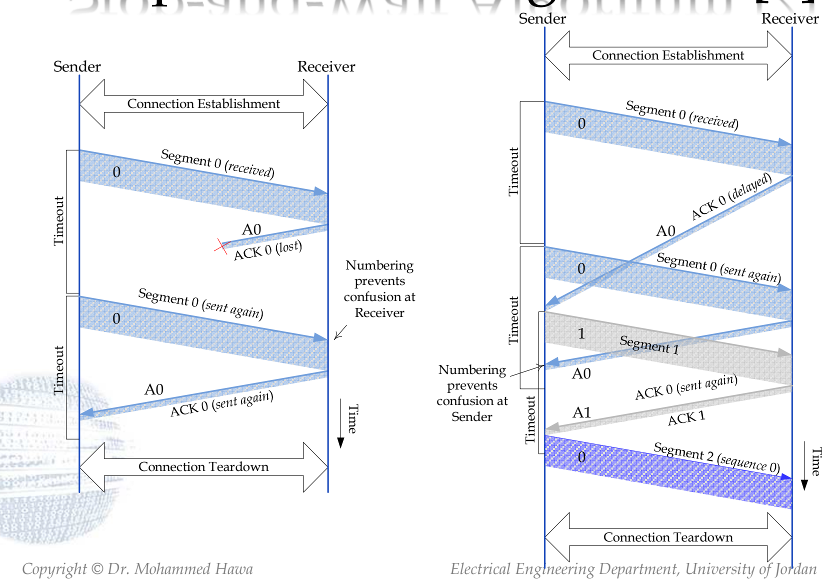
- Automatic Repeat Request (ARQ):
  1. Stop-and-Wait
  2A. Sliding Window – Selective Repeat
  2B. Sliding Window – Go Back N

1. Stop-and-Wait Algorithm, also called Alternating Bit Protocol (ABP):
After transmitting one segment, the sender waits for an acknowledgment before transmitting the next segment. If the acknowledgment does not arrive after a certain period of time, the sender times out and retransmits the original segment.

Stop-and-Wait Algorithm [1]

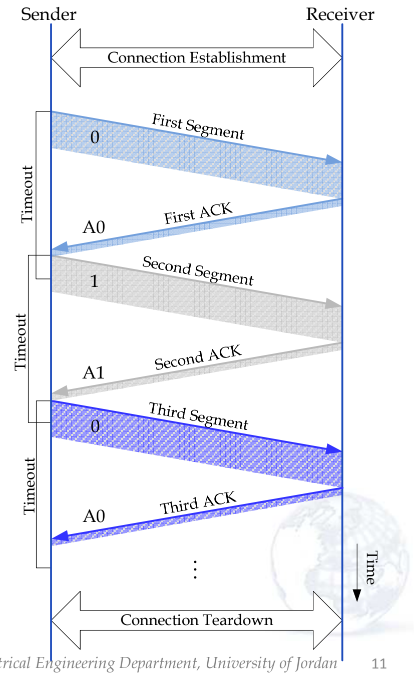


Stop-and-Wait Algorithm [2]



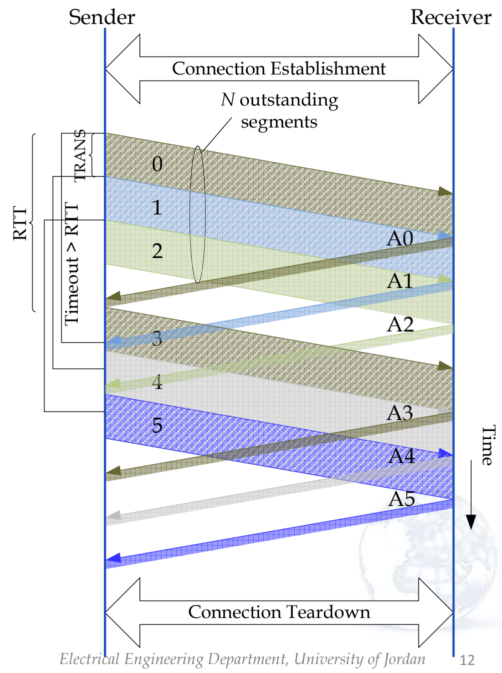
Stop-and-Wait Algorithm [3]

- One bit is enough for sequence number.
- Simple algorithm.
- Wasted time, especially if RTT >> TRANS



Sliding Window

- Transmit more segments while waiting for ACKs.
- More efficient in utilizing link capacity.
- More complex algorithm.



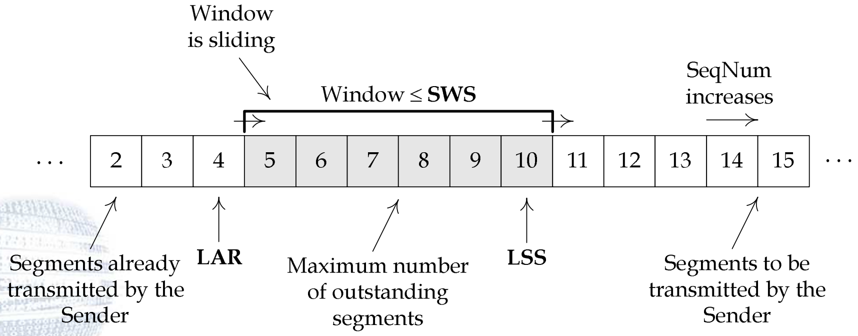
Sliding Window Algorithm - Sender

- In the Sliding Window algorithm, the sender assigns a sequence number, denoted SeqNum, to each segment.
- The sender maintains three variables:
  - **Send Window Size (SWS):** gives the upper bound on the number of outstanding (unacknowledged) segments that the sender can transmit.
  - **Last Acknowledgment Received (LAR):** denotes the sequence number of the last acknowledgment received.
  - **Last Segment Sent (LSS):** denotes the sequence number of the last segment sent.

Sliding Window Algorithm - Sender

- The sender also maintains the following invariant:

$$LSS - LAR \leq SWS$$



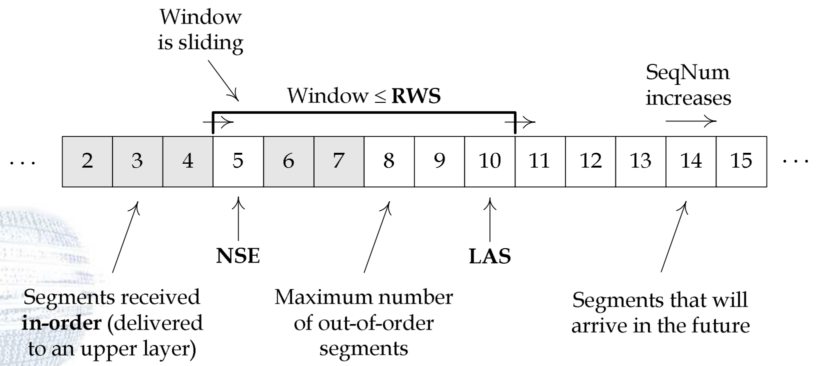
Sliding Window Algorithm - Receiver

- The receiver maintains the following three variables:
  - **Receive Window Size (RWS):** gives the upper bound on the number of out-of-order segments that the receiver is willing to accept.
  - **Next Segment Expected (NSE):** denotes the sequence number of the next segment expected to arrive based on the correct order.
  - **Last Acceptable Segment (LAS):** denotes the sequence number of the largest acceptable out-of-order segment.

Sliding Window Algorithm - Receiver

- The receiver also maintains the following invariant:

$$LAS - NSE + 1 \leq RWS$$



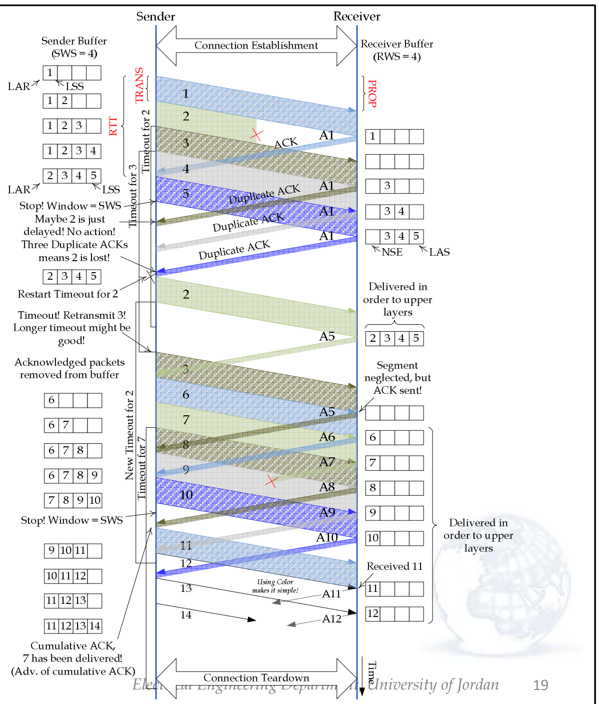
Sliding Window Algorithm

- When a frame with sequence number SeqNum arrives:
  - If SeqNum < NSE or SeqNum > LAS, segment is discarded. An ACK is sent to the sender anyway.
  - If NSE ≤ SeqNum ≤ LAS, segment is accepted. The receiver then sends a cumulative ACK, which acknowledges the largest sequence number not yet acknowledged. Sets:
    NSE = SeqNumToAck + 1,
    LAS = SeqNumToAck + RWS.

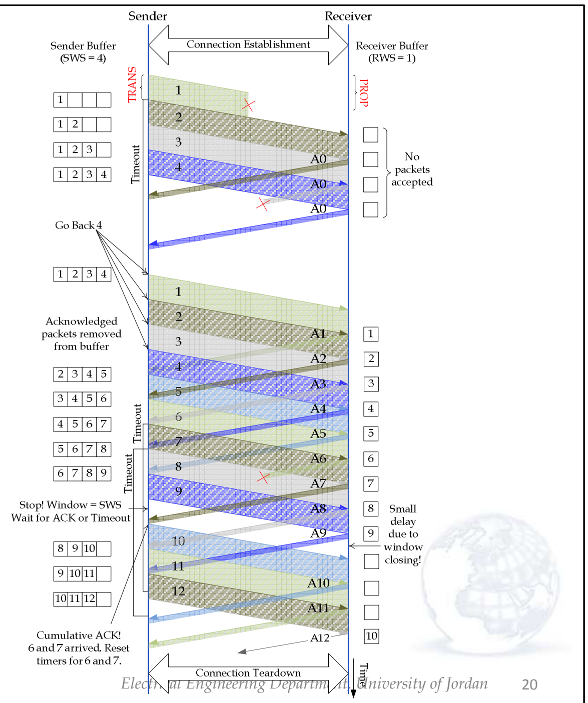
Sliding Window Variants

- **2.A. Selective Repeat:** When a timeout happens (or three duplicate ACKs arrive) indicating that a segment is lost, the sender retransmits that lost segment only. After that, the sender continues to send segments normally (if the SWS allows that). This is the required behavior if RWS = SWS.
- **2.B. Go Back N:** When a timeout happens (or three duplicate ACKs arrive) indicating that a segment is lost, the sender retransmits that lost segment and all the following segments that have already been transmitted before continuing to send segments normally (if the SWS allows that). This is the required behavior if RWS = 1.

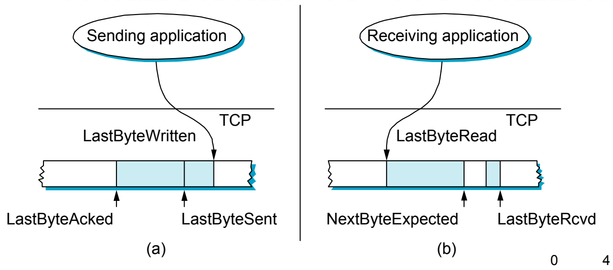
Example "Selective Repeat" Sliding Window

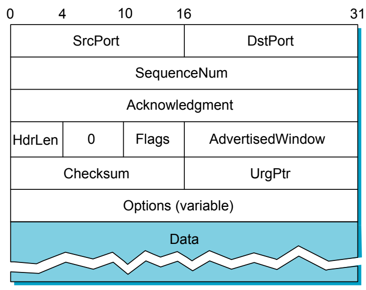


Example "Go Back N" Sliding Window



Remider: TCP Segment Format





Finite Sequence Numbers

- In the Sliding Window Algorithm:
- It turns out that if RWS = 1 (i.e., Go Back N), then we need:

$$\text{MaxSeqNum} \geq \text{SWS} + 1$$

- However, if RWS = SWS (i.e., Selective Repeat), then we need:

$$\text{MaxSeqNum} \geq 2\,\text{SWS} - 1$$

TCP Connection Establishment (establishing a socket, Winsock)