Transport Layer (IV)

TCP Congestion Control (cont.)

- Exercise: Alice wants to send a large file to Bob. The Receiver Window of Bob is 12 KB, the Congestion Window of Alice is 5 KB, and the unacknowledged data is 3 KB.
  - How many additional bytes can she send immediately? [2 KB]
    - The sender's flight-size limit is min(12, 5) = 5 KB. Since 3 KB is already outstanding, Alice can send 2 KB more.
  - If Bob advertises a larger receive window, will Alice necessarily be able to send more? [No]

UDP

- UDP does not have built-in congestion control.
- Many **multimedia applications** (WhatsApp use both TCP and UDP, but prefers UDP for voice transmission,) often **run over UDP**.
- When running over UDP, applications can pump their audio and video into network at a constant rate and occasional loose packets, rather than reduce their rates to “fair” levels at times of congestion and not lose any packet.
- From the perspective of TCP, the multimedia applications running over UDP are not being fair - they do not cooperate with other connections or adjust their transmission rates appropriately.
- Because TCP congestion control will decrease its transmission rate, while UDP sources need not, it is possible for UDP sources to crowd out TCP traffic.

- User Datagram Protocol (RFC 768)
  - A wrapper protocol for IP to add port numbers
  - 8 bytes

Source Port	Destination Port
Length	Checksum

Evolving Transport-layer Functionality

- TCP and UDP are the principal transport protocols for the past 40 years
- Different “flavors” of TCP developed, for specific scenarios. E.g.,
  - Large data transfers (long, fat pipes) face the challenges that many packets “in flight” and loss would shuts down pipeline
  - Wireless networks face the challenges that losses are due to noisy wireless links and mobility and TCP treat this as congestion loss.
- So, moving transport-layer functions to application layer, on top of UDP — HTTP/3: QUIC

- QUIC: Quick UDP Internet Connection

- QUIC implements many of the functions traditionally provided by TCP, but does so on top of UDP, while integrating TLS 1.3 security.
- **QUIC Reliability**:
  - QUIC uses packet numbers, acknowledgements, and retransmission for reliability.
  - Two differences from TCP are: 1) QUIC retransmits lost information in new packets with new packet numbers; 2) acknowledgments are for those received packets.
  - QUIC separates packet numbers from stream byte offsets. Packet numbers identify transmissions. Stream offsets identify where application data belongs. Therefore, it's easier for the sender to measure RTT, and retransmit data in different packet sizes if needed.
- QUIC Congestion Control:
  - It uses mechanism conceptually similar to TCP. It maintains a congestion window, estimates RTT, tracks bytes in flight, detects packet loss, and adjusts its sending rate.

Disclaimer: Notes adapted from the textbook and online resources.