TCP - Introduction

• The Internet Protocol (IP) provides unreliable datagram service between hosts
• The Transmission Control Protocol (TCP) provides reliable data delivery
  – It uses IP for datagram delivery
  – Compensates for loss, delay, duplication and similar problems in Internet components

Features of TCP

• Connection oriented: An application requests a “connection” to destination and uses connection to transfer data
  – IP does not use “connections” - each datagram is sent independently!
• Point-to-point: A TCP connection has two endpoints (no broadcast/multicast)
• Reliability: TCP guarantees that data will be delivered without loss, duplication or transmission errors
Features of TCP (cont’d)

- **Full duplex**: Endpoints can exchange data in both directions simultaneously
- **Reliable connection startup**: TCP guarantees reliable, synchronized startup between endpoints (using “three-way handshake”)
- **Gracefully connection shutdown**: TCP guarantees delivery of all data after endpoint shutdown

Delivering TCP

- **TCP segments travel in IP datagrams**
  
  ![Diagram of TCP segment, IP header, and Payload]

- Internet routers only look at IP header to forward datagrams
- Note: each segment contains a sequence number
**Delivering TCP**

- TCP at destination interprets TCP messages

**TCP and Reliable Delivery**

- TCP provides reliable delivery, recovering from:
  - Lost packets
  - Duplicate packers
  - Delayed packers
  - Corrupted data
  - Transmission speed mismatches
  - Congestion
  - System reboots
Lost Packets

- Recipient sends acknowledgment control message (ACK) to sender to verify successful receipt of data
  - ACKs usually are carried onboard other TCP packets
  - However, even if an application has nothing to transmit, it must transmit acknowledgment packets for each packet it receives
- Thus, for each packet sent, a host expects to receive an acknowledgment, which ensures that the packet did not get lost
  - What if the packet or the acknowledgment get lost?

Lost Packets (cont’d)

- Retransmission timer
  - When a data segment is sent, a timer is started
  - If the segment is acknowledged before the timer expires, the timer is stopped and reset
  - Otherwise, the segment is retransmitted (and the timer is reset and started again)
- The choice of the timeout is critical!
  - If timeout is too long: overall throughput may be reduced (always waiting for acknowledgments)
  - If timeout is too short: too many packets get retransmitted (may increase network congestion)
Lost Packets (cont’d)

• IMPORTANT: packet retransmission (especially if it has to be carried out on an end-to-end basis) significantly increases latency (delay)
  • For real-time video or audio transmission, delay is a more important performance issue than error rate
  • Thus, in many cases it is preferable to forget the error and simply work with the received data stream

Lost Packets - Example
Flow Control

- Flow control is necessary when a computer in the network transmits data too fast for another computer to receive it
  - E.g., a fast server trying to send 1Gb/s data to a small PC
  - Without some form of control, some data will get lost
- Flow control requires some form of feedback from the receiving computer
  - The sender must be able to realize that it is sending data too fast!

TCP Sliding Window

- TCP uses the ACK packets together with the sliding window mechanism
Congestion Control

• When too many packets are present in a part of a network, we have congestion
  – Performance degrades!

• Reasons for congestion:
  – E.g., all of a sudden, streams of packets arrive on 3 or 4 input lines of a router, and they all need the same output line. A queue will build up; if there is insufficient memory, the queue fills it up, and packets will be lost
Congestion (cont’d)

- Network with 1 Mb/s lines and 1000 computers, half of which are trying to transfer files at 100 Kb/s to the other half
  - The total offered traffic exceeds what the network can handle (congestion)
- Congestion collapse:
  - When congestion occurs, packets get dropped
  - Due to packet loss, packets get retransmitted
  - Congestions gets worse and worse!

TCP and Congestion Control

- TCP has a form of congestion control
- Interprets packet loss as an indicator of congestion
  - When it senses packet loss, it slows down the rate of packet transmission
  - When packets are received correctly, sends packets faster
    - Still within the limits of the sliding window
TCP and UDP

- **TCP** provides end-to-end communication. It takes care of **reliable, error-free transfer** of data, and **in-sequence delivery**

- **UDP** has **less overhead** compared to TCP, but **does not guarantee transfers**
  - TCP is preferred to transfer files
  - UDP is preferred to transfer audio/video streams
    - In real-time streaming, we cannot afford the delay consequent to packet retransmission

- Both protocols support **multiplexing**, i.e. they allow several distinct streams of data between two hosts