TCP Overview

- Most widely used transport protocol on the Internet
- Key properties and features
  - Reliable delivery
  - Full duplex
  - Byte stream
  - Multiplexing through “ports”
  - Flow and congestion control
  - Round-trip delay estimation
  - Connection management
- Can be used with IP over many data link layers and many types of networks
TCP Header Format

TCP Control Flags

- **SYN**: Synchronize sequence numbers
- **ACK**: This segment includes an ack (Ack Number field valid)
- **FIN**: Normal termination of a connection
- **RST**: Reset connection because of an error
- **URG**: Urgent Pointer field valid in header
- **ECN, CWR**: Used to support Explicit Congestion Notification (ECN)
- **PSH**: Push flag to notify that data must be sent immediately by the sender, and be delivered to the receiving application immediately on arrival.
**TCP Connection Management**

**Recall:** TCP sender, receiver establish "connection" before exchanging data segments

- Initialize TCP variables:
  - seq. #s
  - Buffers, flow control info (e.g., RcvWindow)

- **Client:** connection initiator
  
  ```java
  Socket clientSocket = new Socket("hostname","port number");
  ```

- **Server:** contacted by client
  
  ```java
  Socket connectionSocket = welcomeSocket.accept();
  ```

**Three way handshake:**

- **Step 1:** client host sends TCP SYN segment to server
  - Specifies initial seq #
  - No data

- **Step 2:** server host receives SYN, replies with SYNACK segment
  - Server allocates buffers
  - Specifies server initial seq. #

- **Step 3:** client receives SYNACK, replies with ACK segment, which may contain data

---

**TCP Connection Management (cont.)**

**Closing a connection:**

- Client closes socket:
  
  ```java
  clientSocket.close();
  ```

- **Step 1:** client end system sends TCP FIN control segment to server

- **Step 2:** server receives FIN, replies with ACK. Closes connection, sends FIN.
**TCP Connection Management (cont.)**

**Step 3:** client receives FIN, replies with ACK.
- Enters "timed wait" - will respond with ACK to received FINs

**Step 4:** server, receives ACK. Connection closed.

**Note:** with small modification, can handle simultaneous FINs.

---

**TCP Connection Management (cont)**

TCP client lifecycle

TCP server lifecycle