TCP/IP protocol stack

TCP/IP Model

Process / Application

Host-to-Host

Internet

Network Access

OSI Model

7. Application

6. Presentation

5. Session

4. Transport

3. Network

2. Datalink

1. Physical

OSI Encapsulation

Segment

Packet

Frame

Bits

RV Department of Computer Science
TCP end-to-end issues

- Connection setup & teardown
- Variable RTT
  - RTT varies b/w end hosts
  - RTT varies during time of day
- Unordered arrivals
- End buffer resource learning & tracking
  - Flow control
- Network congestion
  - Congestion control
3-way handshake

Active Participant (client)

SYN, Seq# = 236

SYN + ACK, Seq# = 45
  Acknowledgment = 236

ACK, Acknowledgment = 46

Passive Participant (server)
Sliding window algorithm

- Reliable & Ordered delivery

<table>
<thead>
<tr>
<th>Sending Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastByteAcked &lt;= LastByteSent</td>
</tr>
<tr>
<td>LastByteSent &lt;= LastByteWritten</td>
</tr>
</tbody>
</table>
Sliding window (contd)

- Receiving side

  LastByteRead < NextByteExpected

  NextByteExpected \leq \text{LastByteReceived} + 1
Sliding window ..

MaxRecvBuffer

LastByteRead  NextByteExpect  LastByteRcvd

Advt Window

LastByteAcked  LastByteSent  LastByteWritten

RV Department of Computer Science
Sliding window (cont’d)

Flow Control

LastByteRcvd – LastByteRead <= MaxRcvBuffer

AdvertisedWindow = MaxRcvBuffer

–{(NextByteExpect -1) – LastByteRead}
Sliding window (contd)

- Flow Control

\[
\text{LastByteSent} - \text{LastByteAcked} \leq \text{AdvertisedWindow}
\]

\[
\text{EffectiveWindow} = \text{AdvertisedWindow} - (\text{LastByteSent} - \text{LastByteAcked})
\]
Adaptive Retransmission

- $\text{EstRTT} = \alpha \times \text{EstRTT} + (1 - \alpha) \times \text{SampleRTT}$
- $\text{TimeOut} = 2 \times \text{EstRTT}$

- Karn/Partridge Algorithm
  - Do’t take sampleRTT when segment is retransmitted; double timeout value if segment is retransmitted

- Jacobson/Karels Algorithm
  - Read
Additive Increase/ Multiplicative Decrease

Slow Start
  – Increase congestion window by 1 for each ack received (why slow start?)

Fast Retransmit & Fast Recovery
  – Read