Outline

- The Transport Layer
- The TCP Protocol
  - TCP Characteristics
  - TCP Connection setup
  - TCP Segments
  - TCP Sequence Numbers
  - TCP Sliding Window
  - Timeouts and Retransmission
  - (Congestion Control and Avoidance)
- The UDP Protocol
The Transport Layer

- What is the transport layer for?
- What characteristics might it have?
  - Reliable delivery
  - Flow control
  - ...

Review of the transport layer

Nick (Leland.Stanford.edu) - Dave (Athena.MIT.edu)

Application Layer

Transport Layer

Network Layer

Link Layer
Layering: The OSI Model

Layering: Our FTP Example
TCP Characteristics

- TCP is connection-oriented.
  - 3-way handshake used for connection setup.
- TCP provides a stream-of-bytes service.
- TCP is reliable:
  - Acknowledgements indicate delivery of data.
  - Checksums are used to detect corrupted data.
  - Sequence numbers detect missing, or mis-sequenced data.
  - Corrupted data is retransmitted after a timeout.
  - Mis-sequenced data is re-sequenced.
  - (Window-based) Flow control prevents over-run of receiver.
- TCP uses congestion control to share network capacity among users. We'll study this in the next lecture.

TCP is connection-oriented

<table>
<thead>
<tr>
<th>(Active)</th>
<th>(Passive)</th>
<th>(Active)</th>
<th>(Passive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Server</td>
<td>Client</td>
<td>Server</td>
</tr>
<tr>
<td>Syn</td>
<td>Syn + Ack</td>
<td>Fin</td>
<td>(Data +) Ack</td>
</tr>
<tr>
<td>Ack</td>
<td></td>
<td>Fin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Connection Setup</td>
<td>Connection Close/Teardown</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3-way handshake</td>
<td>2 x 2-way handshake</td>
<td></td>
</tr>
</tbody>
</table>
TCP supports a “stream of bytes” service

...which is emulated using TCP “segments”
The TCP Segment Format

- IP Data
- TCP Data
- TCP Header
- IP Header

TCP Header and Data = IP Addresses

TCP Header:
- Src port
- Dist port
- Sequence #
- Ack Stream #
- Window
- Checksum
- Urg Pointer
- Flags
  - URG
  - ACK
  - PSH
  - RST
  - SYN
  - FIN
- Window Size

Source and destination port numbers and IP addresses uniquely identify the socket.

Sequence Numbers

Host A: ISN (initial sequence number)

Sequence number = 1st byte

Host B

Ack sequence number = next expected byte
Initial Sequence Numbers

Connection Setup
3-way handshake

TCP Sliding Window

- How much data can a TCP sender have outstanding in the network?
- How much data should TCP retransmit when an error occurs? Just selectively repeat the missing data?
- How does the TCP sender avoid over-running the receiver’s buffers?
TCP Sliding Window

- Window is meaningful to the sender.
- Current window size is "advertised" by receiver (usually 4k – 8k Bytes when connection set-up).
- TCP's Retransmission policy is "Go Back N".

TCP Sliding Window

Host A

Host B

ACK

(1) RTT > Window size

(2) RTT = Window size
TCP: Retransmission and Timeouts

TCP uses an adaptive retransmission timeout value:
- Congestion
- RTT changes frequently
- Changes in Routing

Picking the RTO is important:
- Pick a value that's too big and it will wait too long to retransmit a packet.
- Pick a value too small, and it will unnecessarily retransmit packets.

The original algorithm for picking RTO:
1. \( \text{EstimatedRTT}_k = \alpha \text{EstimatedRTT}_{k-1} + (1 - \alpha) \text{SampleRTT} \)
2. \( \text{RTO} = 2 \times \text{EstimatedRTT} \)

Characteristics of the original algorithm:
- Variance is assumed to be fixed.
- But in practice, variance increases as congestion increases.
TCP: Retransmission and Timeouts

- There will be some (unknown) distribution of RTTs.
- We are trying to estimate an RTO to minimize the probability of a false timeout.

- Router queues grow when there is more traffic, until they become unstable.
- As load grows, variance of delay grows rapidly.

<table>
<thead>
<tr>
<th>Load (Amount of traffic arriving to router)</th>
<th>Average Queueing Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance grows rapidly with load</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load (Amount of traffic arriving to router)</th>
<th>Average Queueing Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance grows rapidly with load</td>
<td></td>
</tr>
</tbody>
</table>

Newer Algorithm includes estimate of variance in RTT:

- Difference = SampleRTT - EstimatedRTT
- Estimated\(RTT_k = \text{Estimated}\(RTT_{k-1} + (\delta \times \text{Difference})
- Deviation = \text{Deviation} + \delta( |\text{Difference}| - \text{Deviation})

- \(RTO = \mu \times \text{EstimatedRTT} + \phi \times \text{Deviation}
  \mu = 1
  \phi = 4

Same as before
TCP: Retransmission and Timeouts

Karn’s Algorithm

Problem:
How can we estimate RTT when packets are retransmitted?

Solution:
On retransmission, don’t update estimated RTT (and double RTO).

User Datagram Protocol (UDP)

Characteristics

- UDP is a connectionless datagram service.
  - There is no connection establishment; packets may show up at any time.
- UDP packets are self-contained.
- UDP is unreliable:
  - No acknowledgements to indicate delivery of data.
  - Checksums cover the header, and only optionally cover the data.
  - Contains no mechanism to detect missing or mis-sequenced packets.
  - No mechanism for automatic retransmission.
  - No mechanism for flow control, and so can over-run the receiver.
User-Datagram Protocol (UDP)

Packet format

- SRC port
- DST port
- checksum
- length
- DATA

Why do we have UDP?
- It is used by applications that don’t need reliable delivery, or
- Applications that have their own special needs, such as streaming of real-time audio/video.