Connection establishment?
Connectionless communication?
Congestion control?
Differentiated services?
Duplicate packet detection?
Flow control?
Loss recovery?
Message or record boundaries?
Ordered data delivery?
Out-of-order data delivery?
Quality of service guarantees?
Urgent data indication?

Where and Why Is TCP Used?

TCP Reliability Mechanism

TCP Summary

HTTP://merlot.usc.edu/cs551-712
Bill Cheney
Basic TCP Mechanisms
CS551
Detecting Half-open Connections

Connection Establishment

TCP Mechanisms

Error Recovery is not part of the spec

Connection Setup

TCP Header

Window flow control
Connection tear-down
Send window section
Connection establishment

TCP Authentication

<table>
<thead>
<tr>
<th>Seq</th>
<th>Ack</th>
<th>Len</th>
<th>Flags</th>
<th>Options (variable)</th>
<th>Padding</th>
<th>Checksum</th>
<th>Urgent Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source port | Destination port

A and B must agree on initial sequence number selection:
A and B must agree on initial sequence number selection:
A and B must agree on initial sequence number selection:
Packet loss, unnecessary retransmissions

Problem: Fast sender can overrun receiver

TCP's solution:
Sender limited to a window's worth of unacknowledged data

Flow control different from congestion control
Window sizes passed in every packet to avoid sender

Accurate round-trip time estimation
Needed for high-bandwidth delay connections

Flow control different from congestion control
TCP's solution:
Sender transmits at a rate controlled by the receiver
Window sizes are passed in every packet to avoid sender

Problems: Fast sender can overrun receiver

Accurate round-trip time estimation
Needed for high-bandwidth delay connections

Protection from sequence number wrapping

Implemented using TCP options

Loss of data

Window size limitations

Packet loss, unnecessary retransmissions

Flow control:
Sender sends at a rate controlled by the receiver
Window sizes are passed in every packet to avoid sender

Problems: Fast sender can overrun receiver

Accurate round-trip time estimation
Needed for high-bandwidth delay connections

Protection from sequence number wrapping

Implemented using TCP options
Recall that MSL is 2 minutes in TCP:

- 1.5Mbps (T1): 6.4 hours
- 10Mbps (Ethernet): 57 minutes
- 45Mbps (T3): 13 minutes
- 100Mbps (FDDI): 6 minutes
- 622Mbps (STS-12): 55 seconds
- 1.2Gbps (STS-24): 28 seconds

Protection From Wraparound

Use timestamp (32-bit) to distinguish sequence number.

Recall that AdvertisedWindow field is 16-bit long.

Scaling factor exchanged during connection set-up:

- Recall that the advertised window field is 16-bit long.

Large Windows

Delay × Bandwidth vs. Link Speed, assuming 100ms RTT.

- AdvertisedWindow field = 1.2Gbps (STS-24): 28 seconds
- AdvertisedWindow field = 622Mbps (STS-12): 55 seconds
- AdvertisedWindow field = 155Mbps (STS-3): 4 minutes
- AdvertisedWindow field = 100Mbps (FDDI): 6 minutes
- AdvertisedWindow field = 45Mbps (T3): 13 minutes
- AdvertisedWindow field = 10Mbps (Ethernet): 122KB
- AdvertisedWindow field = 1.5Mbps (T1): 18KB

Delay × Bandwidth vs. Link Speed, assuming 100ms RTT.

- AdvertisedWindow field = 1.2Gbps (STS-24): 14.8MB
- AdvertisedWindow field = 622Mbps (STS-12): 7.4MB
- AdvertisedWindow field = 155Mbps (STS-3): 1.8MB
- AdvertisedWindow field = 100Mbps (FDDI): 1.2MB
- AdvertisedWindow field = 45Mbps (T3): 549KB
- AdvertisedWindow field = 10Mbps (Ethernet): 1.2MB
- AdvertisedWindow field = 1.5Mbps (T1): 122KB

Recall that the advertised window field is 16-bit long.