TCP -- Basic

- TCP was designed to provide a reliable end-to-end byte stream over an unreliable internetwork.
- TCP must furnish the reliability that most users want and that IP does not provide.
- All TCP connections are full-duplex and point-to-point.
  - TCP does not support multicasting and broadcasting.
  - The traffic can go in both directions at the same time.

TCP -- Basic (Continued)

- Large data is broken into segments and sent separately.
- Segments can arrive in any order.
- Discards duplicates.
- Provides flow control.
**Transport Layer**

- Services to the Session Layer
- Error control
- Flow control
- Sequencing

**Creation of TCP Services**

- Both the sender and receiver create end points (sockets)
- The socket has a number (address)
  - IP address
  - 16-bit number local to the host (port)
  - A port is the TCP name for a TSAP

**Ports**

- Well-known ports
  - Port numbers below 256 are for standard services
  - FTP: port 21
  - TELNET: port 23
  - HTTP (World Wide Web): port 80

**TCP Packet Encapsulation**

- Add Port Numbers, packet Sequence Numbers, Acknowledgement Numbers and other fields.
- Form a TCP/IP "stream" - a connection established using handshake and error detection/control through positive acknowledgement.

**TCP Header Format**

- Source port
- Destination port
- Sequence number
- Acknowledgement number
- TCP Header Length: 6 bits
Pseudoheader

<table>
<thead>
<tr>
<th>Source address</th>
<th>Destination address</th>
</tr>
</thead>
<tbody>
<tr>
<td>80000000</td>
<td>Protocol 6</td>
</tr>
</tbody>
</table>

TCP Check Sum

- Use 1's complement
- Add every 16 bits
- Include TCP header, TCP data, and pseudo-header

TCP Segment

- Remember the maximal length of IP packet is 65,535
- The maximal data carried in a TCP segment is 65,535 - 20 (IP header length) - 20 (TCP header length)
- TCP segments without any data are legal and are commonly used for acknowledgments and control messages

Three-way Handshake

- A sends a SYN message to B
  - I'd like to set up a connection and I will start with sequence number s
- B Replies with a SYN and ACK message to A
  - Yes I will talk to you.
- A sends an ACK message to B along with the first piece of data
  - I got your ACK so here's the start of my data.

TCP Connection

Connection Establishment in TCP

Host 1

SYN(SEQ=x)
SYN(SEQ=x, ACK=x+1)
SYN(SEQ=y, ACK=x+1)
SYN(SEQ=x+1, ACK=y+1)
SYN(SEQ=y, ACK=x+1)
SYN(SEQ=x, ACK=y+1)

Host 2

SYN(SEQ=x)
SYN(SEQ=x+1, ACK=y+1)
SYN(SEQ=y, ACK=x+1)
SYN(SEQ=x+1, ACK=y+1)
SYN(SEQ=x, ACK=y+1)
SYN(SEQ=y, ACK=x+1)
SYN(SEQ=x, ACK=y+1)

A

Upper Layer Protocol

1. Passive open
2. Open
3. Active open
4. Open ID
5. SYN(SEQ=55)
6. SYN(SEQ=202, ACK=56)
7. SYN(SEQ=56, ACK=203)
8. Open ID
9. Open Success

B

TCP
Data Transfer in TCP

10. Send 30 bytes
11. DATA (SEQ=57, ACK=203)
12. Deliver 100 bytes
13. Send 100 bytes
14. DATA (SEQ=203, ACK=87)
15. Deliver 150 bytes
16. ACK (SEQ=87, ACK=303)

Graceful Close in TCP

17. Close
18. Send 150 bytes
19. FIN (SEQ=88, ACK=303)
20. DATA (SEQ=303, ACK=89)
21. Closing
22. Close
23. Deliver 150 bytes
24. FIN (SEQ=453, ACK=89)
25. FIN (SEQ=49, ACK=454)
26. Terminate
27. Terminate

User Datagram Protocol (UDP)

- Adds Port Numbers to IP addresses
- A Port number refers to a specific application running on a host. e.g. SMTP uses Port 25 while Telnet uses Port 23.
- Result is a set of Packets

UDP (Continued)

- No handshaking or error control
- Also called a "Connectionless" protocol
- Often referred to as "Unreliable" - meaning error control can't be relied upon.
- Useful for the following situations
  - Overhead is a concern
  - Small data requests such as queries

UDP Header

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source port</td>
<td>16</td>
</tr>
<tr>
<td>Destination port</td>
<td>16</td>
</tr>
<tr>
<td>UDP length</td>
<td>2</td>
</tr>
<tr>
<td>UDP checksum</td>
<td>2</td>
</tr>
</tbody>
</table>

UDP Check Sum

- Use 1's complement
- Add every 16 bits
- Include UDP header, UDP data, and pseudo-header
- 0000000000000000 for no checksum