Transport Layer

COMP476
Networked Computer Systems

Transport Layer

- The first, or lowest, end to end layer.
- The transport layer delivers packets to the appropriate application on the host.
- Some transport layer protocols correct transmission errors.
- There are several popular transport layer protocols including TCP and UDP.

Intermediate Nodes

Internet Port Numbers

- Applications are identified by a 16 bit integer number known as a port number.
- Internet ports do NOT refer to plugs in the back of the machine.
- The full address of an application is InternetName:port
- Applications bind to a port number to receive data sent to that port.

Well Known Ports

- Port numbers under 2K are reserved for specific “well known” application servers
  - 21 ftp
  - 23 telnet
  - 79 finger
  - 80 HTTP web servers
  - 443 HTTPS secure web servers
  - 17 Quote of the Day

Lesser Known Ports

- Well Known Ports are only used by servers.
- Servers for non-standard applications use higher numbered ports.
- Applications accessing a server use a higher numbered port.
- When a program connects to a remote system, it is automatically assigned a port.
Ports in a URL

- Some web servers, particularly test systems, use a port other than 80.
- The port number follows a colon after the IP name


Multiple Protocols

- There are two major transport layer protocols in the Internet Protocol suite
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
- The different protocols provide different services.

TCP and UDP

<table>
<thead>
<tr>
<th>TCP</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Oriented</td>
<td>Connectionless</td>
</tr>
<tr>
<td>Complete reliability</td>
<td>best effort delivery</td>
</tr>
<tr>
<td>corrects lost, corrupted and out-of-order packets</td>
<td>Full Duplex communication</td>
</tr>
<tr>
<td>Full Duplex communication</td>
<td>Point to Point communication</td>
</tr>
<tr>
<td>Point to Point communication</td>
<td>to 1, many to many</td>
</tr>
<tr>
<td>Stream Interface</td>
<td>Message Oriented</td>
</tr>
<tr>
<td>Reliable connection startup</td>
<td>no connection</td>
</tr>
</tbody>
</table>

Popular TCP Applications

- HTTP - web protocol
- telnet - terminal protocol
- ftp - file transfer protocol
- any program that has lengthy transfers that require reliability.

Popular UDP Applications

- DNS requests
- WINS requests
- Streaming Audio
- Any application that needs to send a short amount of data that can be resent if necessary (*idempotent or at-least-once*).
- Time critical applications

Connection Oriented

- A program using TCP must connect to the remote host before sending any data.
- Connection verifies ability to communicate with the destination.
- To establish a connection, TCP sends:
  - message to the server requesting a connection.
  - Response from server accepting connection.
  - Message to the server specifying parameters.
Error Correction
- TCP corrects lost, corrupted, delayed and out-of-order packets.
- Applications do not have to worry about the transmission reliability.
- Data is delivered exactly as it was sent.
- UDP does not correct any errors. Higher level software must recover from problems.
- UDP is far more efficient than TCP.

Point to Point Communications
- TCP connects one sender to one receiver.
- All data sent over a TCP connection goes to the same receiving application.
- TCP connections do not support broadcasts.
- Connections are full duplex allowing communication in both directions.

Arbitrary Communications
- UDP does not require a connection.
- A program using UDP can send or receive messages to and from any other system.
- A program can broadcast messages using the IP address 255.255.255.255
- UDP programs can receive broadcasts.

Packet Interface
- UDP sent: 50 bytes
- UDP receive 50 bytes
- UDP receive 25 bytes: truncated
- UDP receive 75 bytes: Received less than request

Stream Interface
- TCP sent: 50 bytes
- TCP receive 50 bytes
- TCP receive 25 bytes: truncated
- TCP receive 75 bytes: Received less than request

TCP Header
- Source Port: 16 bits
- Destination Port: 16 bits
- Sequence Number: 32 bits
- Acknowledgement Number: 32 bits
- Payload Offset: 4 bits
- Flags: 6 bits
- Window: 16 bits
- Checksum: 16 bits
- Urgent Pointer: 16 bits
- Options (if any): 16 bits
- Beginning of Data: 1 bit
**UDP Header**

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Destination Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Length</td>
<td>Checksum</td>
</tr>
</tbody>
</table>

Data

The checksum is optional. It can be set to zero to omit error checking.