Lecture 6 (Transport Layer Protocol)

1 Transport Services and Protocol

1. Provide logical communication between application processes running on different hosts
2. It is only present at the end nodes and intermediate nodes implement upto network layer (one level below transport layer)
3. Sender breaks application messages into segments and passes it to the network layer
4. Receiver reassembles these segments into messages and then passes it to the application layer
5. Two transport protocols are available: TCP and UDP

1.1 TCP

1. reliable, in-order delivery
2. congestion control
3. flow control
4. connection setup

1.2 UDP

1. unreliable, unordered delivery
2. no-frills extension of “best effort” IP

Note: Transport layer doesn’t provide delay and bandwidth guarantee

2 Multiplexing and Demultiplexing

Notation:

1. **Message:** the entire application layer content
2. **Segment:** the “chunks” transport header + message is divided into
3. **Datagram:** network header + transport header + message

2.1 Multiplexing

Data from multiple sockets is handled by adding suitable transport header

2.2 De-Multiplexing

Use header info from transport layer to deliver received segments to correct socket

2.2.1 How De-Multiplexing Works

1. Datagram has source and destination IP address
2. Each datagram carries one transport layer segment
3. Each segment has source, destination port number
4. Host uses this IP address and port number to direct segment to the correct socket

Port number 0 to 1023 are *reserved* port numbers and applications need to make sockets on different port numbers, from 1024 to 65,535

2.2.2 Connectionless De-Multiplexing

UDP datagrams with same destination port number but different source IP addresses and/or port numbers will be redirected to same socket at receiving host

2.2.3 Connection Oriented De-Multiplexing

TCP socket is identified by a 4-tuple:

1. source IP
2. source port
3. destination IP
4. destination port

3 UDP

1. UDP segments may be lost or delivered out of order
2. The protocol is connectionless and each segment is independent of every other
3. Even with the drawbacks, it is used since it doesn't have any congestion control and hence can be fast even with congestion, and UDP is *simple*
4. UDP is used by streaming multimedia apps, DNS, SNMP (Simple Network Management Protocol), HTTP/3 (reliability and congestion control is added at application layer)
5. The header is made up of 64 bits: source port number, destination port number, length of the message, checksum