COMP 3331/COMP 9331 Week 4

Transport Layer Part 1

**Key concepts in this lecture**

* multiplexing, demultiplexing
* reliable data transfer
* flow control
* congestion control
* UDP: connectionless transport
* TCP: connection-oriented reliable transport

# 3.1 transport-layer services

Transport layer provides logical communication between app processes running on different hosts. It runs in end systems. On the sender side, transport layer breaks app msgs into segments and passes them to network layer (The network layer doesn’t guarantee reliable transfer of data). On the receiver side, transport layer reassembles segments into messages, passes to app layer. The transport layer has many-application-processes, every application may use different ports.

Diagram

Description automatically generated

# 3.2 multiplexing and demultiplexing

* **Multiplexing at ender** – handle data from multiple sockets, add transport header
* **Demultiplexing at receiver** – use header info to deliver received segments to correct socket

Diagram, text

Description automatically generated

* **Pay attention to the source IP, port, dest IP, port**
* **All the port at server side use port 80**

Diagram

Description automatically generated

5775

6428

6428

5775

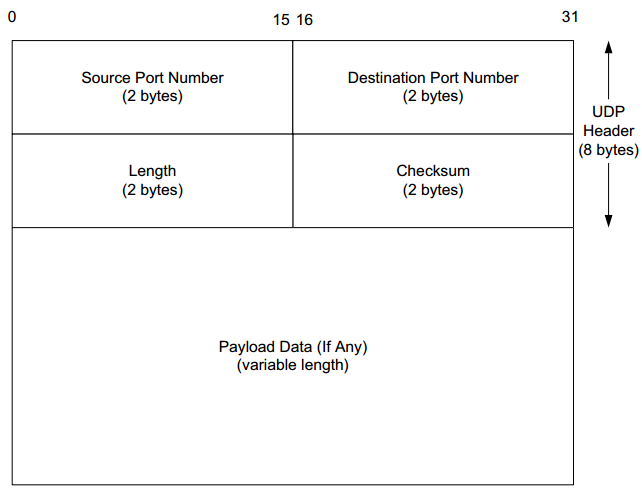
**Pay attention to the source IP, port, dest IP, port**

# 3.3 connectionless transport: UDP

User Datagram Protocol (UDP)

* Cannot handle with reliable transfer service, only multiplexing/demux
* Connectionless – no handshaking, each UDP segment handled independently of each other

## UDP segment header (8 bytes)



* Header is 32 bits wide – source + dest port no. (both 16 bits), length, checksum

## Why do we still use UDP?

* Simple – no connection state at sender/receiver
* Small header size
* Very useful in video streaming where latency requirement is high, no congestion control, UDP can blast away as fast as desired

## UDP checksum

It can only detect 1 bit error.

**Sender:**

* treat segment contents, including header fields, as sequence of 16-bit integers
* checksum: addition (one’ s complement sum) of segment contents
* sender puts checksum value into UDP checksum field

**Receiver:**

* Add all the received together as 16-bit integers
* Add that to the checksum
* If the result is not 1111 1111 1111 1111, there are errors

Text

Description automatically generated

Note: The leading 1 is wrapped around to the end of the number as +1

## Application of UDP:

* Quick request/response (DNS, DHCP)
* Network management (SNMP)
* Routing updates (RIP)
* Voice/video chat
* Gaming (especially FPS)

# 3.4 principles of reliable data transfer

Things we have to deal with in data transfer

* a packet is corrupted
* a packet is lost (router, switches use certain buffers that are overloaded, packets loss)
* a packet is delayed (queued in buffer)
* packets are reordered (internet doesn't guarantee every packet takes the same path)
* packet is duplicated (retransmission if packet is lost, but the original packet is actually not lost, we end up having duplicate packet)

# reliable data transfer (very important topic)

**Rdt 1.0: reliable transfer over a reliable channel – underlying channel perfectly reliable**

• no bit errors

• no loss of packets

Graphical user interface

Description automatically generated with medium confidence

**rdt2.0: channel with bit errors**

underlying channel may flip bits in packet

Diagram

Description automatically generated