

#5: Transport Layer: UDP and TCP

User Datagram Protocol - Request For Comments RFC 768

1) Adds little service on top of IP

- connectionless multiplexing / de-multiplexing
- checksum

2) transmission is unreliable

- often used by streaming multimedia apps

3) to achieve reliable transmission over UDP

- require error detection and recovery mechanisms

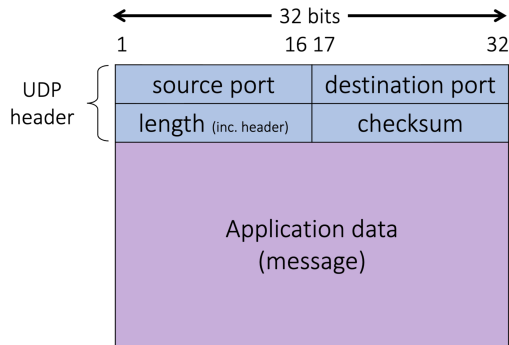
Electronic process to allow more than one electrical signal to be sent using only one connection

Data from multiple sources to be sent using only one transmission channel

When transport layer receives UDP segment

- check destination port # in segment
- Directs UDP segment to the socket with that port #
- IP datagrams with the same destination port # will be directed to the same UDP socket at destination.

UDP segment structure



Transmission Control Protocol

1) Connection Oriented
- handshake

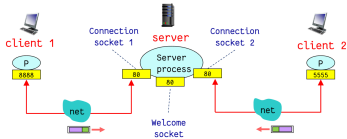
2) Reliable, in-order byte stream
- Application passes data to TCP and TCP forms segments in view of MSS (maximum segment size)

3) Flow control and congestion control
- prevent unnecessary losses

Connection-oriented De-mux

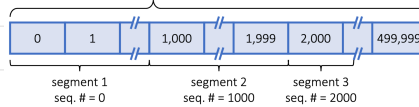
A TCP connection (socket) is identified by 4-tuple:

- (srcIPAddr, srcPort, destIPAddr, destPort)
- Receiver uses all four values to direct a segment to the appropriate socket.



TCP Sequence number

→ byte number of the first byte of data in the segment



TCP Acknowledgement number

↳ seq # of the next byte of data expected

TCP Sender Events (simplified)

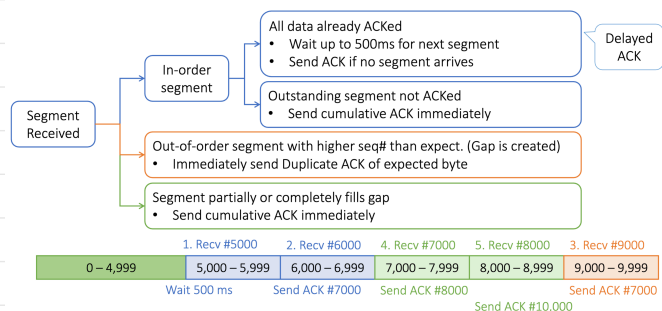
```

Loop(forever)
  switch(event)
  event: data received from application
    create TCP segment with nextSeqNum
    if (timer not currently running) ← Sender only keeps one timer
      start timer
      pass segment to IP
      nextSeqNum += length(data)

  event: timer timeout
    retransmit unacknowledged segment with smallest seq num
    start timer ← Retransmit only oldest unACK segment

  event: ACK received, with ACK num #y
    if (y > sendBase) ← Cumulative ACK
      sendBase = y
      if (there are still unacknowledged segments)
        start timer
    
```

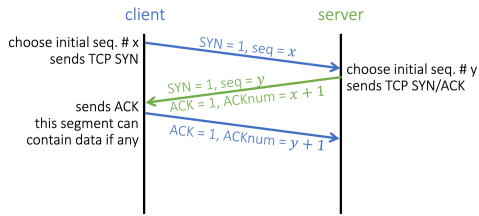
TCP Receiver Events



TCP Connection Establishment

Established using a 3-way handshake

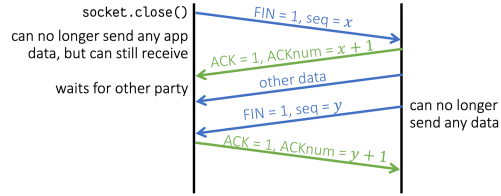
- Agree on connection and exchange parameters



TCP Closing Connection

Each side closes their own side of the connection

- Send segments with FIN bit set
- No more sending of data after FIN, but can still receive



TCP segment structure

