TCP

- Transport Control Protocol

- Design parameters and objectives

  - used by most popular applications, majority of Internet traffic is transported over TCP
  
  - significant impact on congestion behavior of the Internet
  
  - must operate over networks with widely-varying characteristics
  
  - must be robust and (relatively) simple to implement
TCP Header

<table>
<thead>
<tr>
<th>Offsets Octet</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Octet</strong></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td><strong>Bit</strong></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>Source port</td>
<td>Destination port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Sequence number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Acknowledgment number (if ACK set)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Data offset</td>
<td>Reserved</td>
<td>NS</td>
<td>CWR</td>
</tr>
<tr>
<td>16</td>
<td>Checksum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Options (if Data Offset &gt; 5, padded at the end with &quot;0&quot; bytes if necessary)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Another image appropriated from Wikipedia...
TCP Sliding Window

- Initial sequence #
- Ack’d data, delivered to the application
- Received and ack’d but not yet delivered to the application
- Received but not yet ack’d
- Sent but not yet received
- Receiver buffer size
- Receiver buffer available for data to be sent
- Window size
- Moves as data is sent
- Moves as application reads data
- Moves with ACKs send
- ACKs rec’d
SESSION MANAGEMENT

OPEN

CLIENT "OPEN" SERVER

"OK"

"OK"

TCP

SYN=1
ACK=0

SEQ#={X}
ACK#={2}

SYN=1
ACK=1

SEQ#={Y}
X+1

SYN=∅

ACK=1

SEQ#={X+1}

ACK#={Y+1}

3-WAY HANDSHAKE

SYN - PROPOSING SEQ#

ACK - ACK# FIELD CONT

VALID VALUE
TCP Flow Control

- **Receiver congestion**
  - **Window Size field** - explicitly reported by the receiver
  - **TCP Window Scale Option**

- **Network congestion**
  - **Retransmission timeout** - based on observed RTT
  - **Transmission window** - based on detected packet loss
RECEIVER CONGESTION CONTROL

WRITE(1000) → OS
  Rec. Window: 2000
  Write: 1000
  Ack: 1000

WRITE(1500) → OS
  Ack Window: 1000
  Write: 1500
  Ack: 1500

WRITE(500) → OS
  Ack Window: 0
  Write: 500
  Ack: 500

OS → READ(1500)
  Window Update: 1500

APP
Retransmission Timeout

**Initialization:**

RTO ← 1 sec

**After the first measurement:**

SRTT ← R

RTTVAR ← R/2

RTO ← SRTT + max (G, K * RTTVAR)

**After subsequent measurements:**

RTTVAR ← (1 - beta) * RTTVAR + beta * |SRTT - R'|

SRTT ← (1 - alpha) * SRTT + alpha * R'

RTO ← SRTT + max (G, K * RTTVAR)

Where:

R - first RTT measurement

R' - subsequent RTT measurement

RTTVAR - RTT variance

SRTT - smoothed RTT estimate

RTO - retransmission timeout

G - clock granularity

**Recommended values:**

alpha=1/8, beta=1/4, K=4

RFC 6298
Network Congestion Control

- No explicit indication of congestion given
- Source observes RTT and packet loss and adjusts transmission rate according to its estimate of the congestion state of the network
- Additive Increase Multiplicative Decrease (AIMD)
  - better safe than sorry