TÉCNICO LISBOA

Lab 6: Transmission Control Protocol (TCP)

Using TCP sockets.
Understanding TCP signaling, reliability, and congestion control

TCP

- Goals:
  1. Implement a simple *TCP client/server* communication using *sockets*
  2. Understand the TCP protocol *handshake* mechanism and explore TCP *packet headers*
  3. Identify the various TCP *congestion control* phases

Evaluation

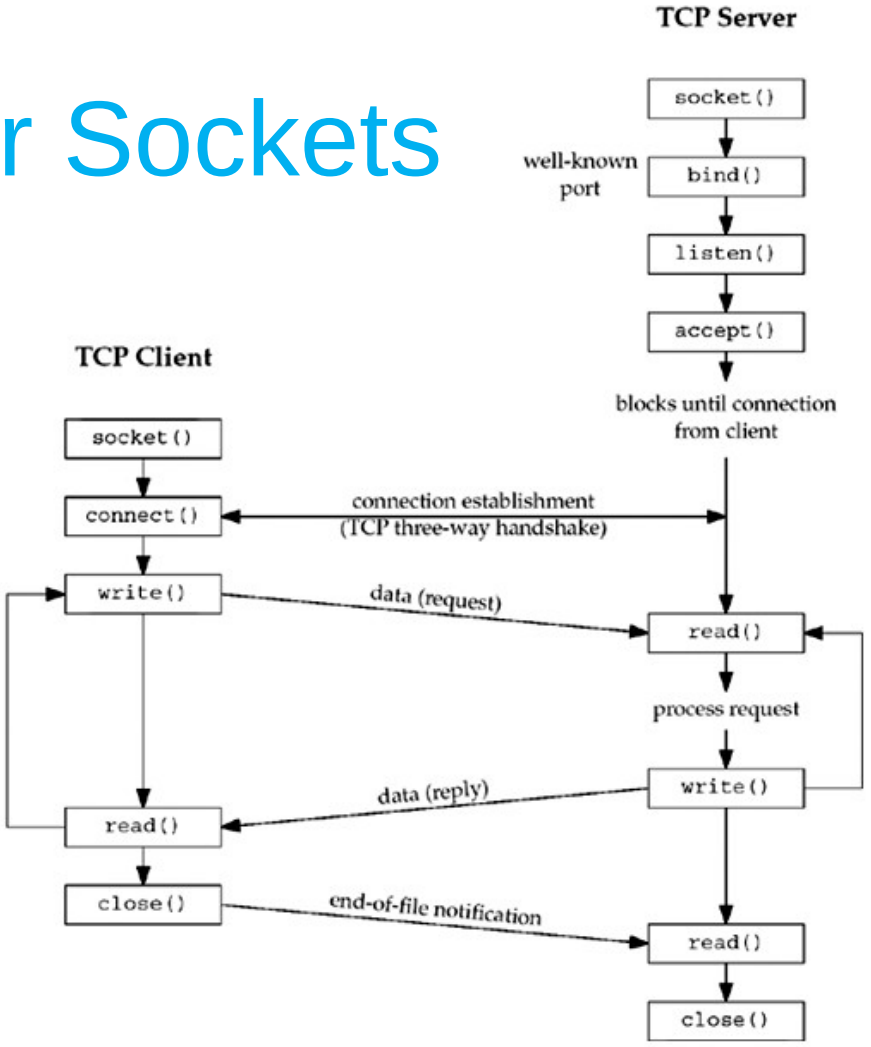
- Where
  - Moodle: *Lab 6: Transmission Control Protocol (TCP)*
- Submission due
  - Sunday, December 3, 23h59

The TCP Protocol

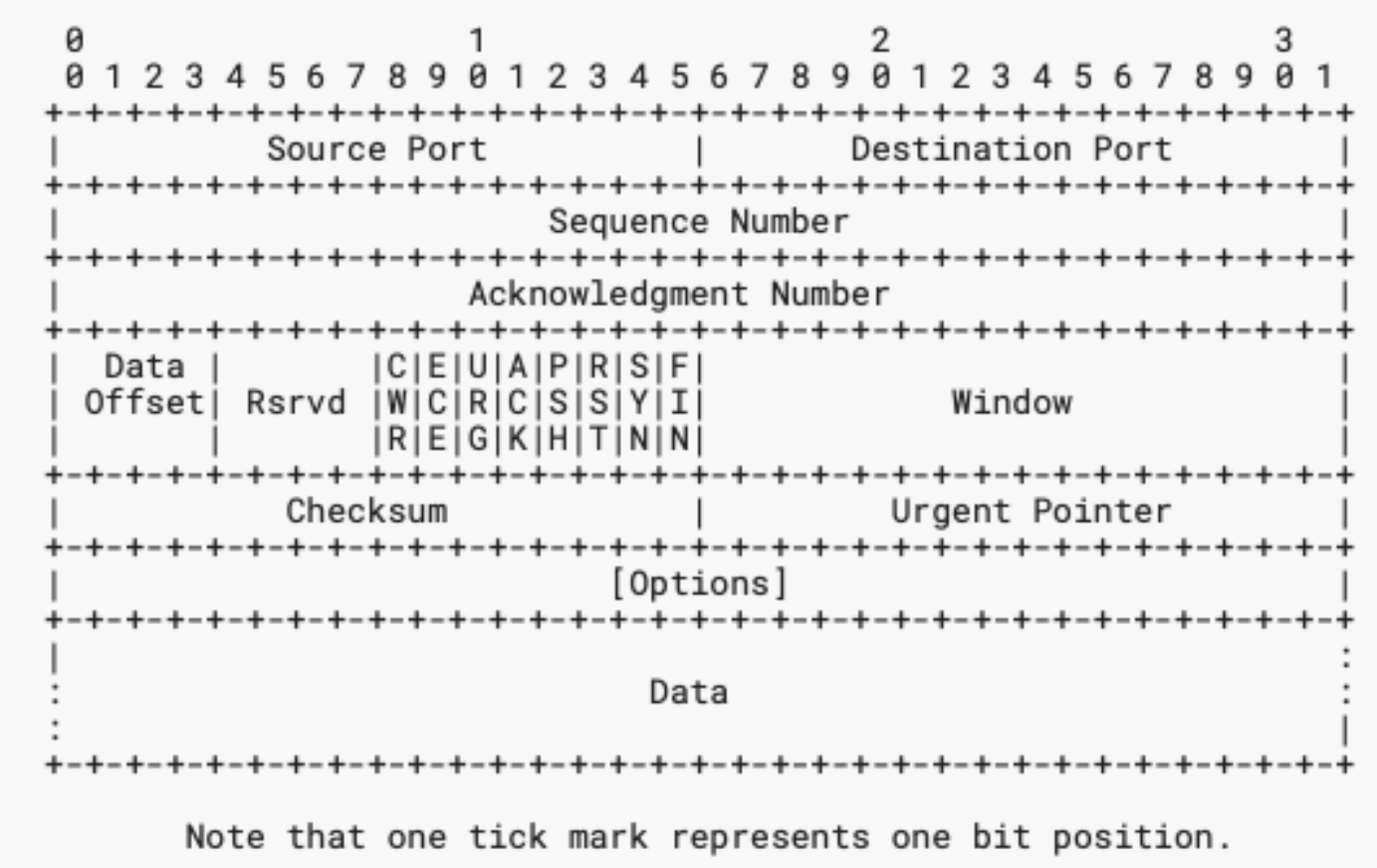
- Transport layer protocol
- Connection oriented service
- Reliable data transfer
- Flow control
- Congestion control

TCP Client & Server Sockets

- Creation
  - Server starts first & waits
  - Client connects to server
- Data Exchange
  - Like a file/pipe (read, write)
- Tear Down



TCP Header


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Source Port
Destination Port
Sequence Number
Acknowledgment Number
Data Offset
Rsrvd
C E U A P R S F
W C R C S S Y I
R E G K H T N N
Window
Checksum
Urgent Pointer
[Options]
Data
Note that one tick mark represents one bit position.

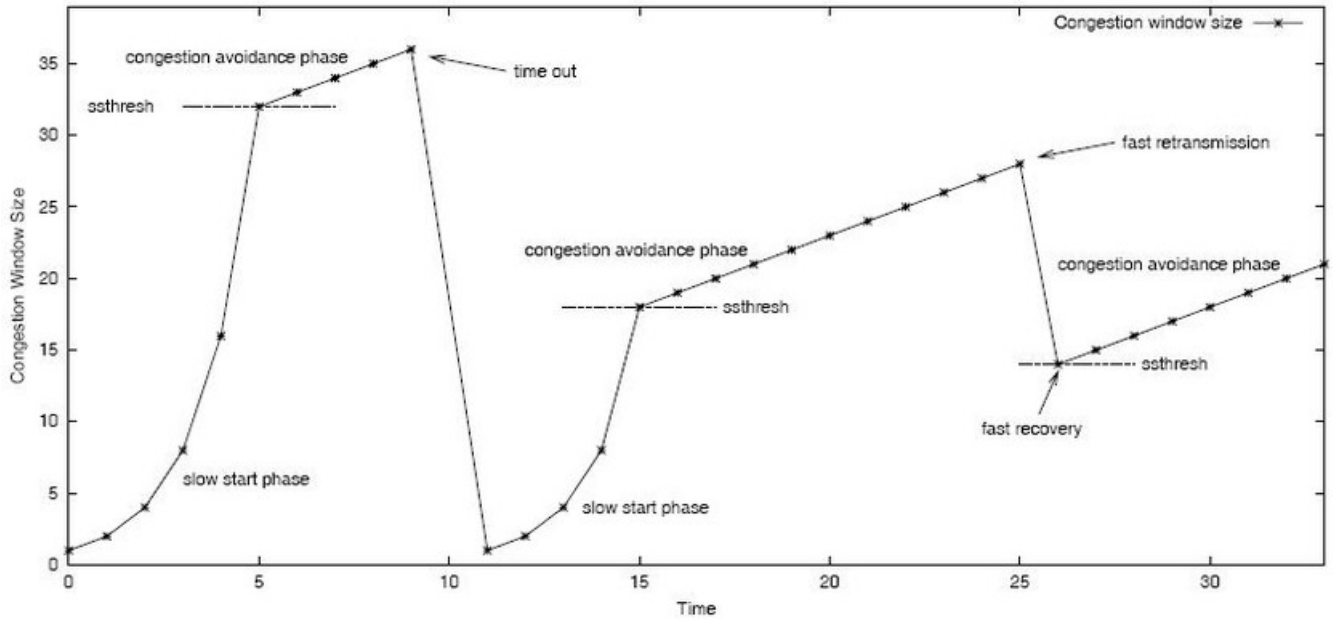
TCP Header - Flags

Flag		Binary	Decimal
CWR	Congestion Window Reduced	1 0 0 0 0 0 0 0	128
ECE	ECN-Echo	0 1 0 0 0 0 0 0	64
URG	Urgent	0 0 1 0 0 0 0 0	32
ACK	Acknowledgement	0 0 0 1 0 0 0 0	16
PSH	Push	0 0 0 0 1 0 0 0	8
RST	Reset	0 0 0 0 0 1 0 0	4
SYN	Syn	0 0 0 0 0 0 1 0	2
FIN	Fin	0 0 0 0 0 0 0 1	1

What is Congestion?

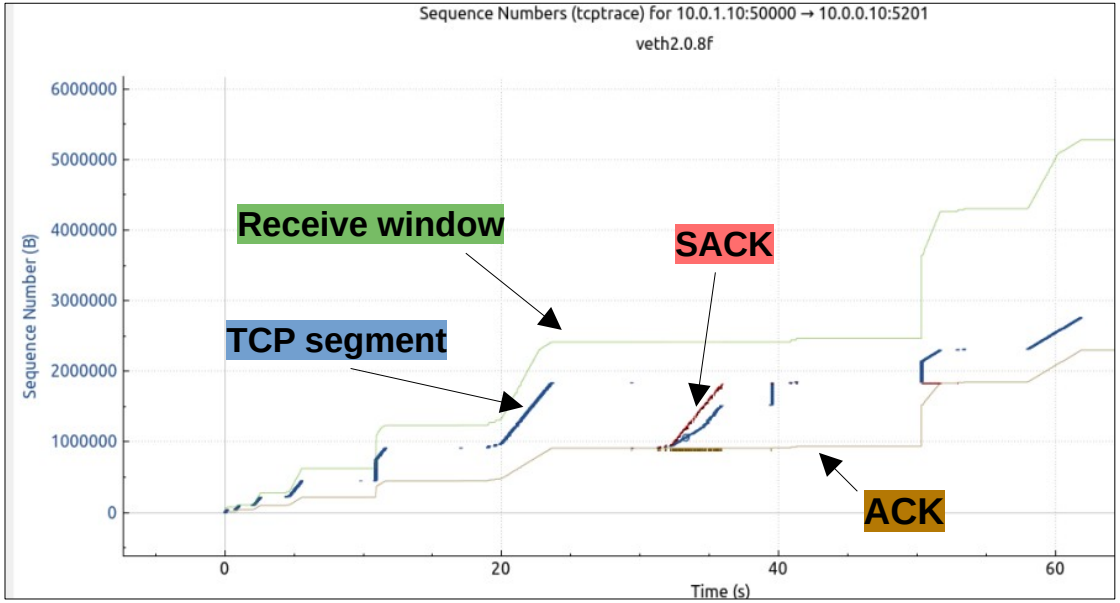
- "*Too many sources* sending *too much* data *too fast* for the network to handle"
- Consequences:
  - Lost packets (buffer overflow at routers)
  - Long delays (queueing in router buffers)

TCP Congestion Control (Reno)



TCP Trace

- Data stream (one direction)
- Sequence number over time



Wireshark → *Statistics* → *TCP Stream Graphs* → *Time Sequence (tcptrace)*