TCP
Transport Control Protocol
DEEPER DIVE
Outline

- Segment format
- Connection management
- Error control
- Flow control
Segment Header

Source Port | Destination Port

Sequence Number

Acknowledgement Number

HLEN | RES | Flags | Receive window

Checksum | Urgent Pointer

Options | Pad

32 bit
Sequence Numbers

- Send 'ABC'
  - Seq=12 data= 'ABC'
  - Seq=34 data= 'XYZ'
  - Seq=15 data= 'DEF'
  - Seq=15 Ack=37

- Rec 'ABC'
  - Seq=34 Ack=15
  - Seq=34 data= 'XYZ'

- Send 'XYZ'
  - Seq=34 data= 'ABC'

- Rec 'DEF'
  - Seq=15 Ack=37
ACK Piggybacking

Send 'ABC'

Seq=12 data='ABC'

Rec 'ABC'

Seq=34 Ack=15 data='XYZ'

Send 'XYZ'

Rec 'DEF'

Seq=15 Ack=37
Flags

→ ACK: Acknowledgement field is valid
→ PSH: This segment requests a push
→ URG: Urgent Pointer is valid
Flags

- SYN: Synchronize sequence number
- Open a connection
- FIN: Sender has reached end of its byte stream
- Close a connection
- RST: Reset connection
CONNECTION MANAGEMENT
Three-way Handshake

App wants to open

SYN Seq=12

SYN ACK Seq=34 Ack=13

OK

ACK Seq=13 Ack=35

App accepts

time
Connection Termination

L

App closes

FIN Seq=12

FIN ACK Seq=34 Ack=13

Seq=34 data=…

Seq=45 data=…

ACK Seq=13 Ack=53

FIN Seq=53

FIN ACK Seq=13 Ack=54

R

OK

App closes

OK
ERROR CONTROL
Retransmission

- Go-back-N
- Segments lost in bursts
- Unnecessary retransmission
- Overload
- Selective retransmit, optional
Double timer for outstanding segments
Time-out Too Short

Round Trip Time estimation

Unnecessary retransmission
Cumulative Acknowledgement

ACK Seq=12 Data=...

ACK Seq=23 Data=...

ACK Seq=34 Ack=23

ACK Seq=34 Ack=31

All well up to 30
Delayed Acknowledgements

Send ‘ABC’
Send ‘DEF’

Seq=12 data= ‘ABC’
Seq=15 data= ‘DEF’

Seq=34 Ack=18

Rec ‘ABC’
Rec ‘DEF’

(timer)
Fast Retransmit

Seq=12
Ack=6

Seq=23
Ack=6

Seq=6
Ack=6

Seq=35

Seq=42
Ack=42

Seq=12

Seq=23

Seq=42
FLOW CONTROL
Sliding Window

Acknowledged  Outstanding  Unsent

ACK

Sliding window

Acknowledged  Outstanding  Unsent
Window Sizing

- Sender buffer
- Store unack’ed bytes
- Receiver
- Flow control
- Network
- Congestion control
On The Receiver Side

→ Provide upper layer with complete ordered stream
→ Buffer
→ Reorder segments
→ Keep data until picked up
Flow Control

- Buffer might become full
- Receiver window advertises available space
- Transmission window always smaller
Congestion Control

→ Network congestion
→ Lost segments
→ Go-back-N retransmission
→ More congestion
→ More lost segments
→ More retransmissions

IN 1986 COMPLETE INTERNET BLOCK
Congestion Control

→ Reduce window size when there is congestion
→ Duplicate ack
→ Time-out
→ Increase unit: MSS
→ Maximum Segment Size
Congestion Control

- Increase when there is no congestion
- Start small (1 MSS) but increase fast (exponential)
- Slow start
- Continue slowly
- Congestion avoidance
Duplicate Ack

→ Moderate congestion
→ Windows is halved
→ Multiplicative decrease
→ Increase of 1 MSS each received window
→ Additive increase
Time-out

⇒ Severe congestion
⇒ Window is set to 1 MSS
⇒ Increase of 1 MSS each received segment
⇒ Exponential increase
Time-out

- Until size is half of window at time-out
- Then additive increase
To Start

- Slow start
- Window = 1 MSS
- Exponential increase
- Until first loss event
- Additive increase
Congestion Control

- Many issues, several variants
- None perfect
- Especially on long fat pipes
- Long reaction time
Saw Tooth Behavior
Synchronization of Connections

Segments from multiple connections discarded during congestion

They all slow down

Then all increase

Until congestion

And all slow down ...