



15-441 Computer Networking

Lecture 18 – More TCP & Congestion Control

Good Ideas So Far...



- Flow control
 - Stop & wait
 - Parallel stop & wait
 - Sliding window (e.g., advertised windows)
- Loss recovery
 - Timeouts
 - Acknowledgment-driven recovery (selective repeat or cumulative acknowledgment)
- Congestion control
 - AIMD → fairness and efficiency
- How does TCP actually implement these?

18.30.2007

Lecture 18: TCP Details

3

Outline



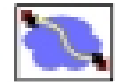
- THE SPOOKY PARTS of TCP
 - If it doesn't solve you here... it will at the final!
- TCP connection setup/data transfer
 - The Handshake Protocol (TCP)
- TCP reliability
 - How to recover your DEAD packets
- TCP congestion avoidance
 - Avoiding the death-trap of overloaded routers

18.30.2007

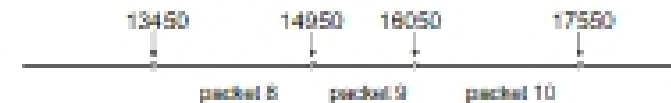
Lecture 18: TCP Details

3

Sequence Number Space



- Each byte in byte stream is numbered.
 - 32 bit value
 - Wraps around
 - Initial values selected at start up time
- TCP breaks up the byte stream into packets.
 - Packet size is limited to the Maximum Segment Size
- Each packet has a sequence number.
 - Indicates where it fits in the byte stream



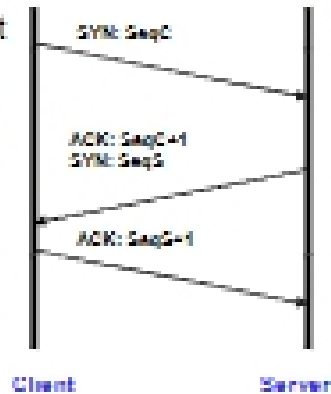
18.30.2007

Lecture 18: TCP Details

4

Establishing Connection: Three-Way handshake

- Each side notifies other of starting sequence number it will use for sending
 - Why not simply chose 0?
 - Must avoid overlap with earlier incarnation
 - Security issues
- Each side acknowledges other's sequence number
 - SYN-ACK: Acknowledge sequence number + 1
- Can combine second SYN with first ACK



10.30.2007

Lecture 18: TCP Details

6

TCP Connection Setup Example

```

08.08.00.040000 IP 100.0.0.0.100.0000 > 100.000.000.00.00: [
  3028000004-3028000004(0) win 00000 ->max 0280,seq,seq,seq000<
  (DF)

08.08.00.100000 IP 100.000.000.00.00 > 100.0.0.0.100.0000: [
  3028000000-3028000000(0) win 0010000000 win 0000 ->max
  0480,seq,seq,seq000< (DF)

08.08.00.100000 IP 100.0.0.0.100.0000 > 100.000.000.00.00: . win
  3028000010 min 00000 (DF)
    
```

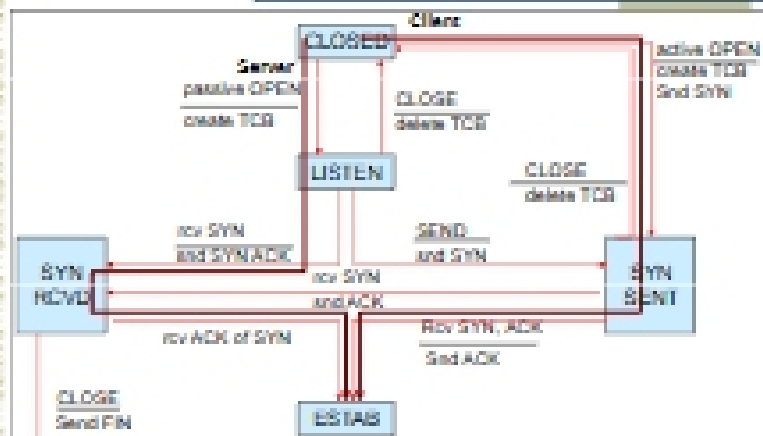
- Client SYN
 - SeqC: Seq. #4010802004, window 85505, max. seq. 1280
- Server SYN-ACK+SYN
 - Receiver: #4010802005 (= SeqC+1)
 - SeqS: Seq. #0428051560, window 5040, max. seq. 1460
- Client SYN-ACK
 - Receiver: #0428051570 (= SeqS+1)

10.30.2007

Lecture 18: TCP Details

8

TCP State Diagram: Connection Setup



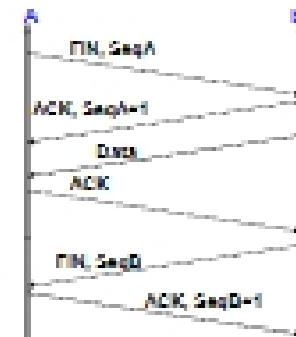
10.30.2007

Lecture 18: TCP Details

7

Tearing Down Connection

- Either side can initiate tear down
 - Send FIN signal
 - "I'm not going to send any more data"
- Other side can continue sending data
 - Half open connection
 - Must continue to acknowledge
- Acknowledging FIN
 - Acknowledge last sequence number + 1



10.30.2007

Lecture 18: TCP Details

9

TCP Connection Teardown Example



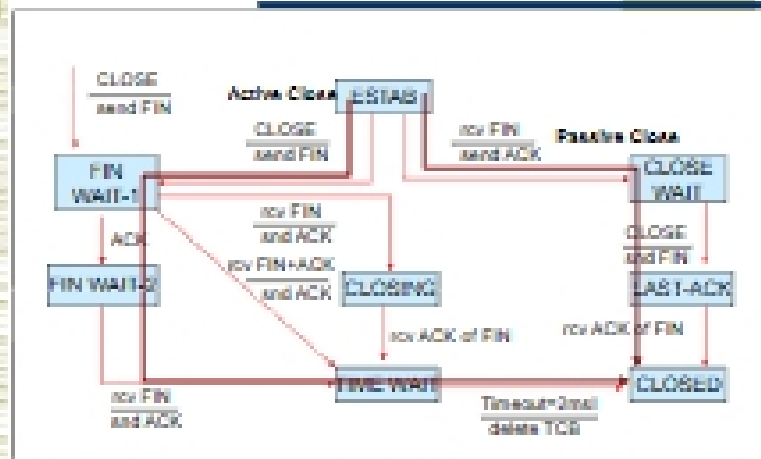
```
08:24:17.000000 IP 128.2.222.198 > 128.2.210.194: P
1480294581:1480294581(0) seq 1480294581 win 0000 (DF)

08:24:17.000700 IP 128.2.210.194 > 128.2.222.198: P
1480787689:1480787689(0) seq 1480787689 win 0000 (DF)

08:24:17.000700 IP 128.2.222.198 > 128.2.210.194: . seq
1480787690 win 0000 (DF)
```

- Session
 - Echo client on 128.2.222.198, server on 128.2.210.194
- Client FIN
 - Seq#: 1480294581
- Server ACK + FIN
 - Ack: 1480294582 (= Seq#+1)
 - Seq#: 1480787689
- Client ACK
 - Ack: 1480787690 (= Seq#+1)

State Diagram: Connection Tear-down



Outline



- TCP connection setup/data transfer
- **TCP reliability**
- TCP congestion avoidance

Reliability Challenges



- Congestion related losses
- Variable packet delays
 - What should the timeout be?
- Reordering of packets
 - How to tell the difference between a delayed packet and a lost one?