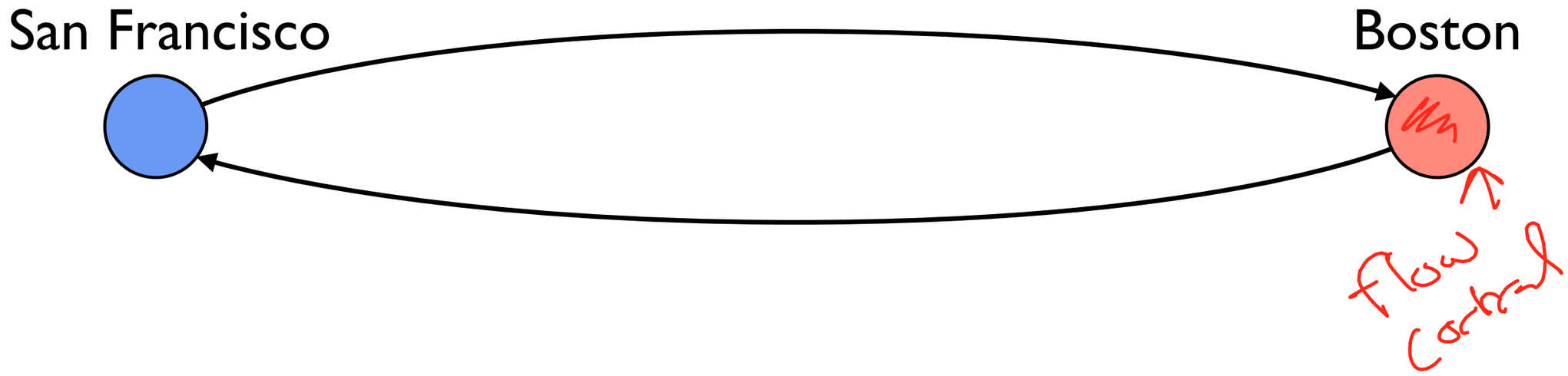
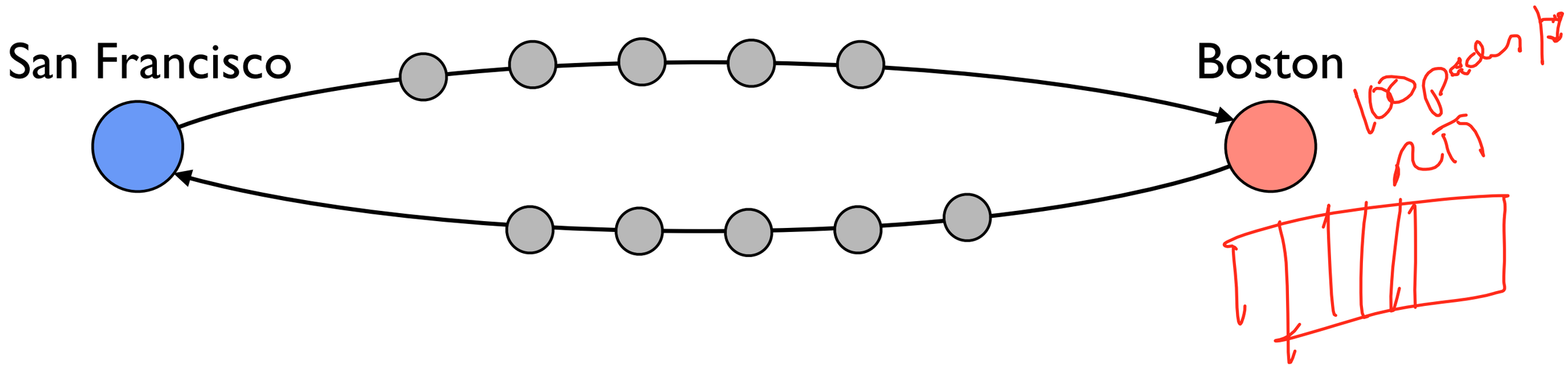
Congestion Control

Congestion, TCP Tahoe, slow start

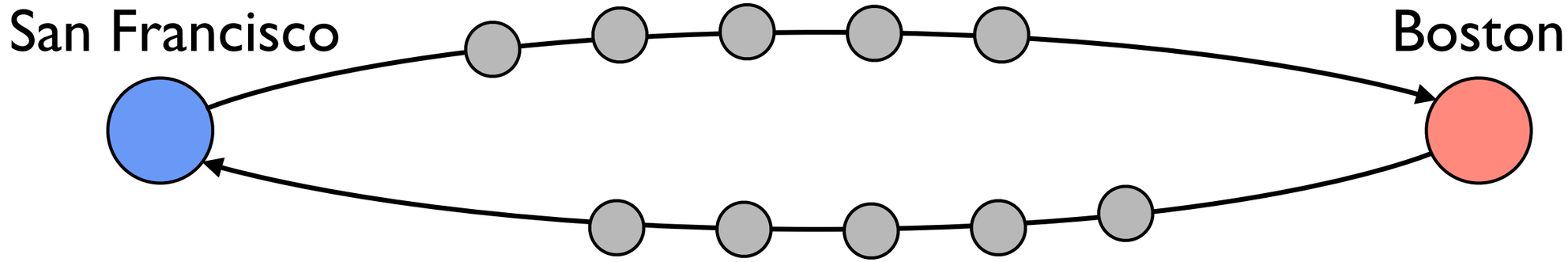
Congestion Control Motivation



Congestion Control Motivation



Congestion Control Motivation



Congestion control: control data rate so it does not congest network, improves overall performance

TCP History

- 1974: 3-way handshake
- 1978: TCP and IP split into TCP/IP
- 1983: January 1, ARPAnet switches to TCP/IP
- 1986: Internet begins to suffer congestion collapse
- 1987-8: Van Jacobson fixes TCP, publishes seminal TCP paper (Tahoe)
- 1990: Fast recovery and fast retransmit added (Reno)

Three Questions

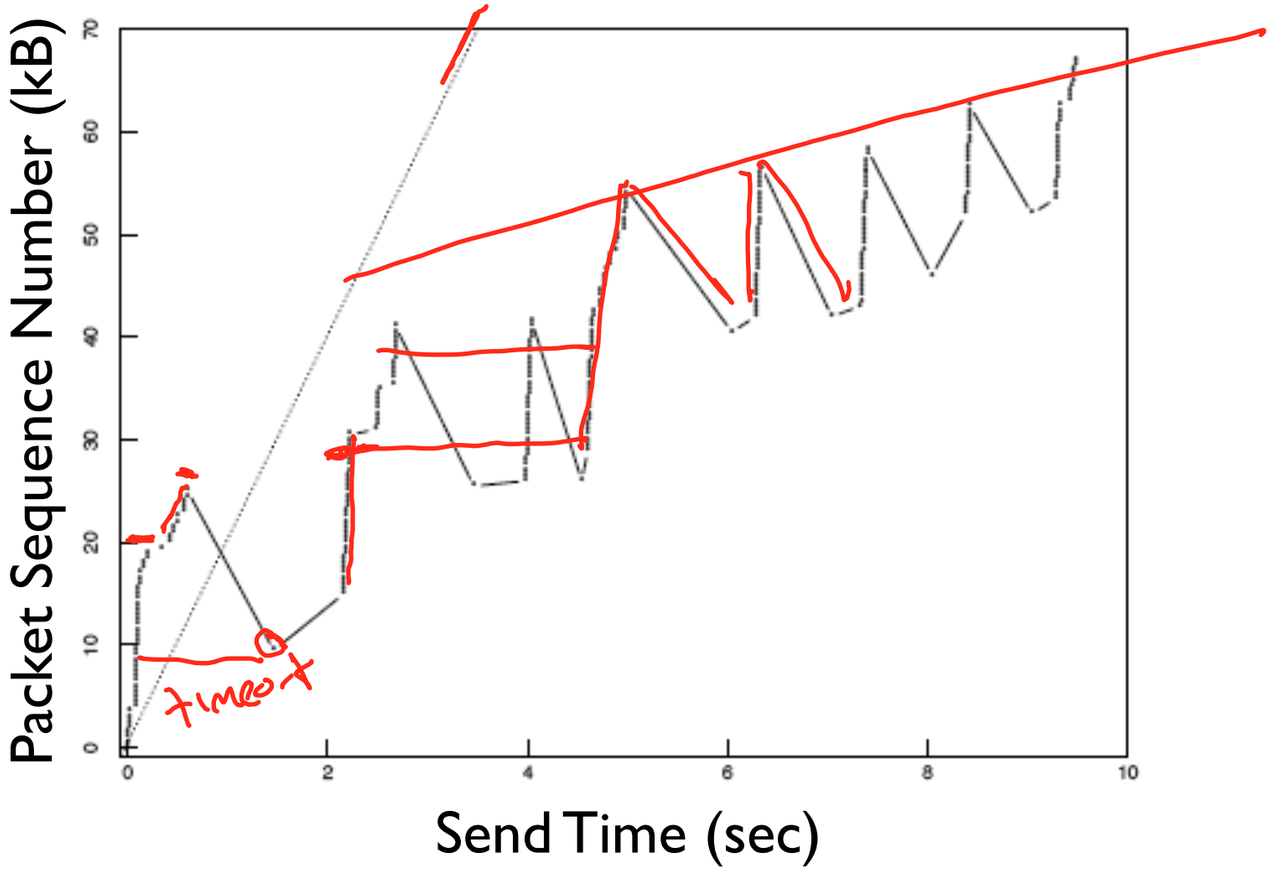
- When should you send new data?
- When should you send data retransmissions?
- When should you send acknowledgments?

TCP Pre-Tahoe

- Endpoint has the flow control window size
- On connection establishment, send a full window of packets
- Start a retransmit timer for each packet
- Problem: what if window is much larger than what network can support?

window 30 KB

TCP in 1986


Packet Sequence Number (kB)
0
10
20
30
40
50
60
70
0
2
4
6
8
10
Send Time (sec)
timeout

Three Improvements

- Congestion window
- Timeout estimation
- Self-clocking

Three Improvements

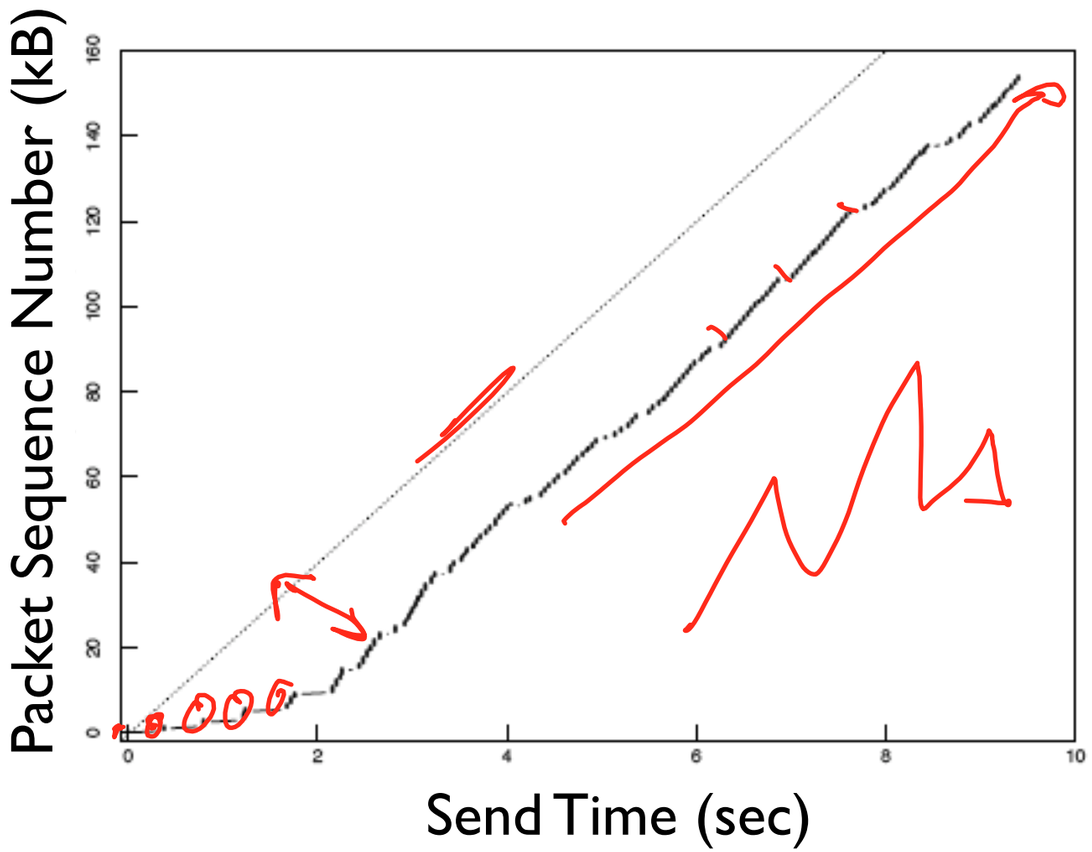
- Congestion window
- Timeout estimation
- Self-clocking

Congestion Window (TCP Tahoe)

- Flow control window is only about endpoint
- Have TCP estimate a *congestion window* for the network
- Sender window = min(flow window, congestion window)
- Separate congestion control into two states
  - Slow start: on connection startup or packet timeout
  - Congestion avoidance: steady operation

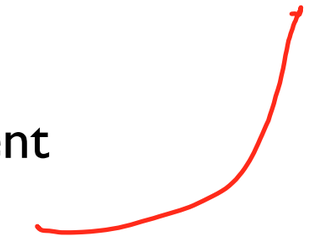
Slow Start Benefits

- Slow start
  - Window starts at Maximum Segment Size (MSS)
  - Increase window by MSS for each acknowledged packet
- Exponentially grow congestion window to sense network capacity
- "Slow" compared to prior approach

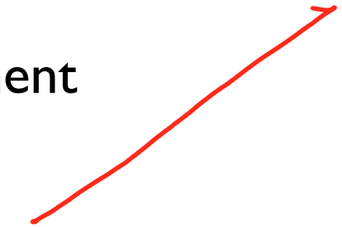


Congestion Avoidance

- Slow start
  - Increase congestion window by MSS for each acknowledgment
  - Exponential increase



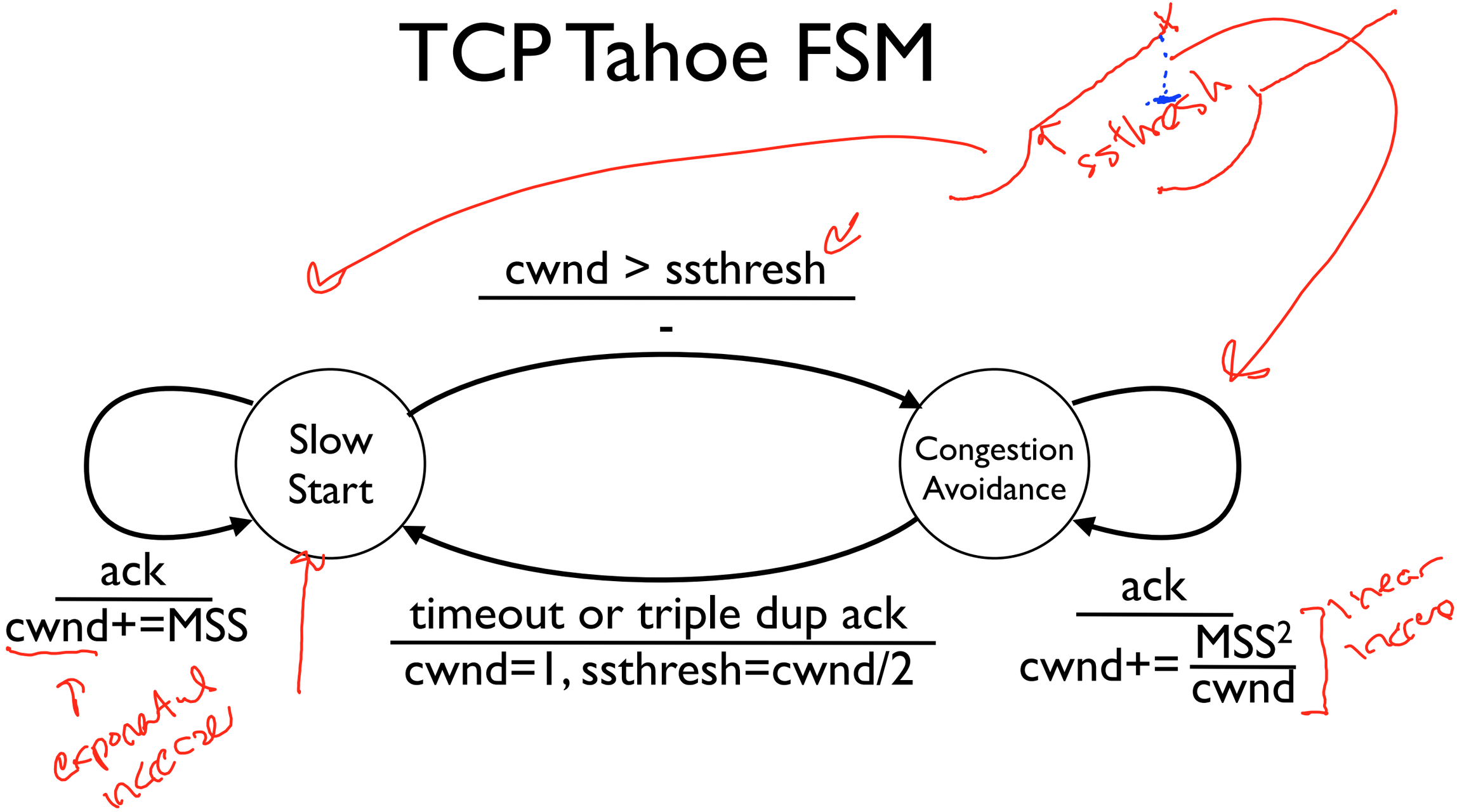
- Congestion avoidance
  - Increase by MSS^2/congestion window for each acknowledgment
  - Behavior: increase by MSS each round trip time
  - Linear increase



State Transitions

- Two goals
  - Use slow start to quickly find network capacity
  - When close to capacity, use congestion avoidance to very carefully probe
- Three signals
  - Increasing acknowledgments: transfer is going well
  - Duplicate acknowledgments: something was lost/delayed
  - Timeout: something is very wrong

TCP Tahoe FSM



TCP Tahoe Walkthrough