BRAC UNIVERSITY

Inspiring Excellence

Transport Layer (TCP Congestion Control)

Lecture 6 | CSE421 – Computer Networks

Department of Computer Science and Engineering
School of Data & Science

Congestion Control Vs Flow Control

- Congestion control try to make sure subnet can carry offered traffic, a global issue involving all the hosts and routers.
  - It can be open-loop based or involving feedback
- Flow control is related to point-to-point traffic between given sender and receiver.
  - it always involves direct feedback from receiver to sender

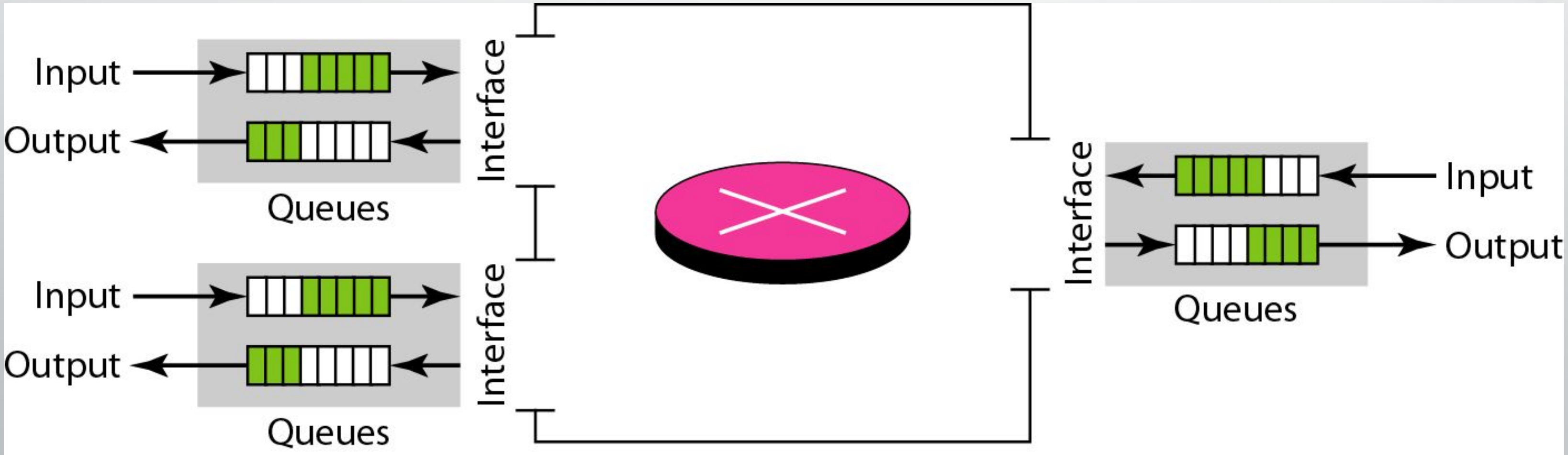
Congestion:

- Congestion occurs

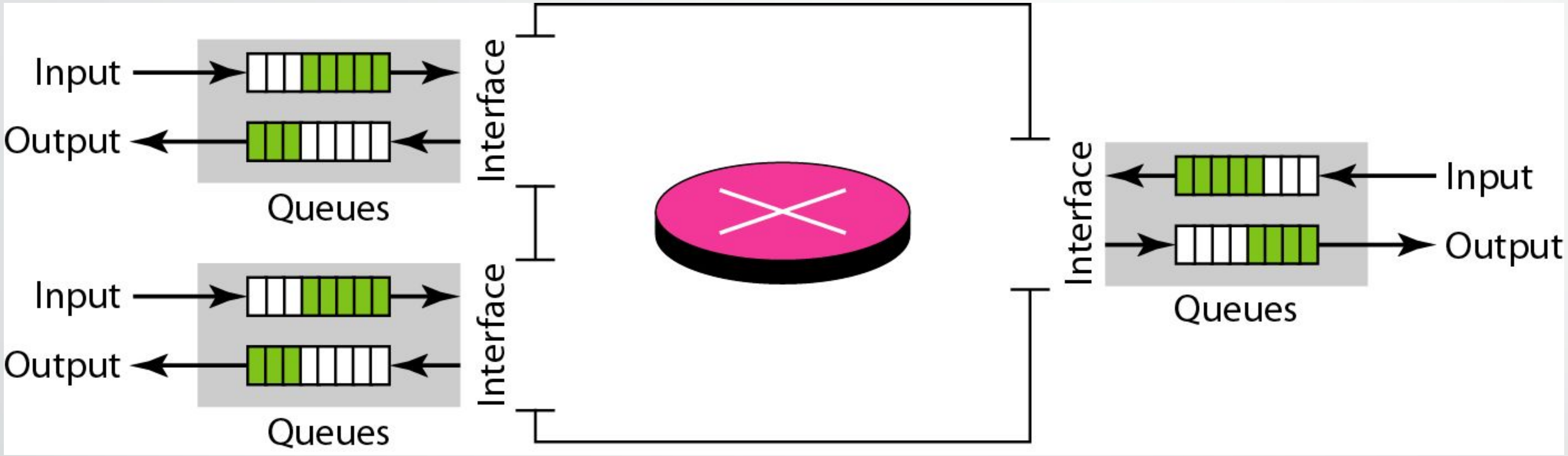
the load on the network > the capacity of the network

the number of packets a network can handle.

the number of packets sent to the network



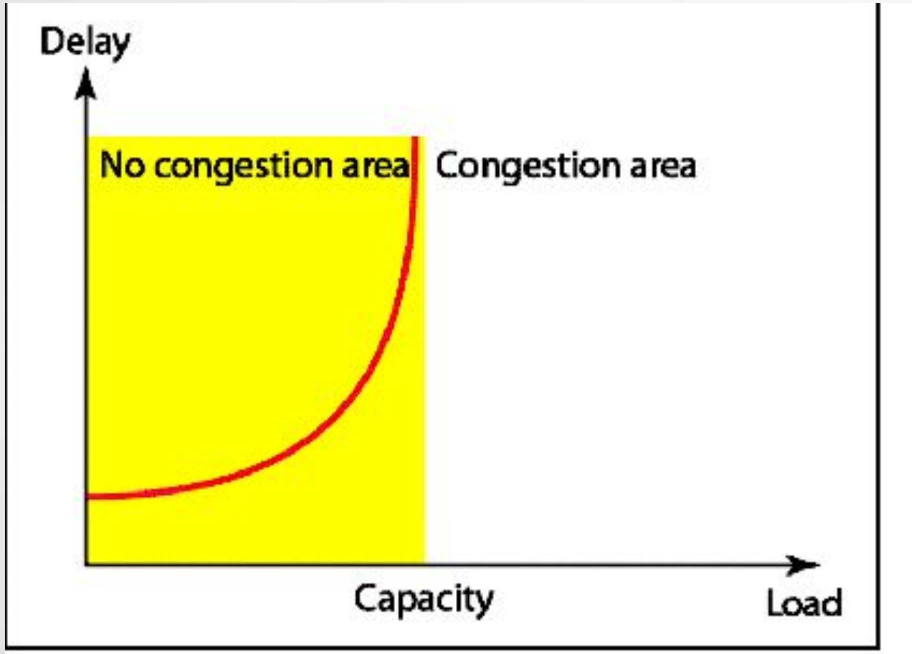
Queues in a router



- If packet arrival rate > the packet processing rate
- input queues becomes longer and longer

- If packet departure rate < the packet processing rate
- output queues becomes longer and longer

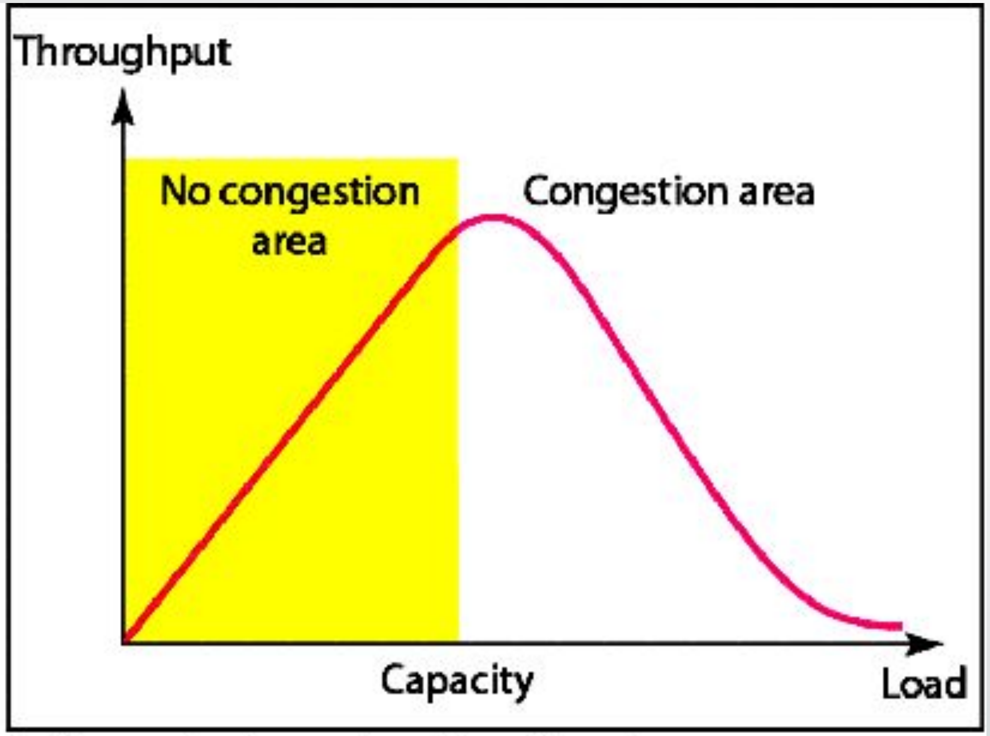
Network Performance



a. Delay as a function of load

- Delay has a negative effect on the load consequently the congestion.
- When a packet is delayed, no ack for source, so source retransmits, making the delay and congestion worse.

Network Performance



b. Throughput as a function of load

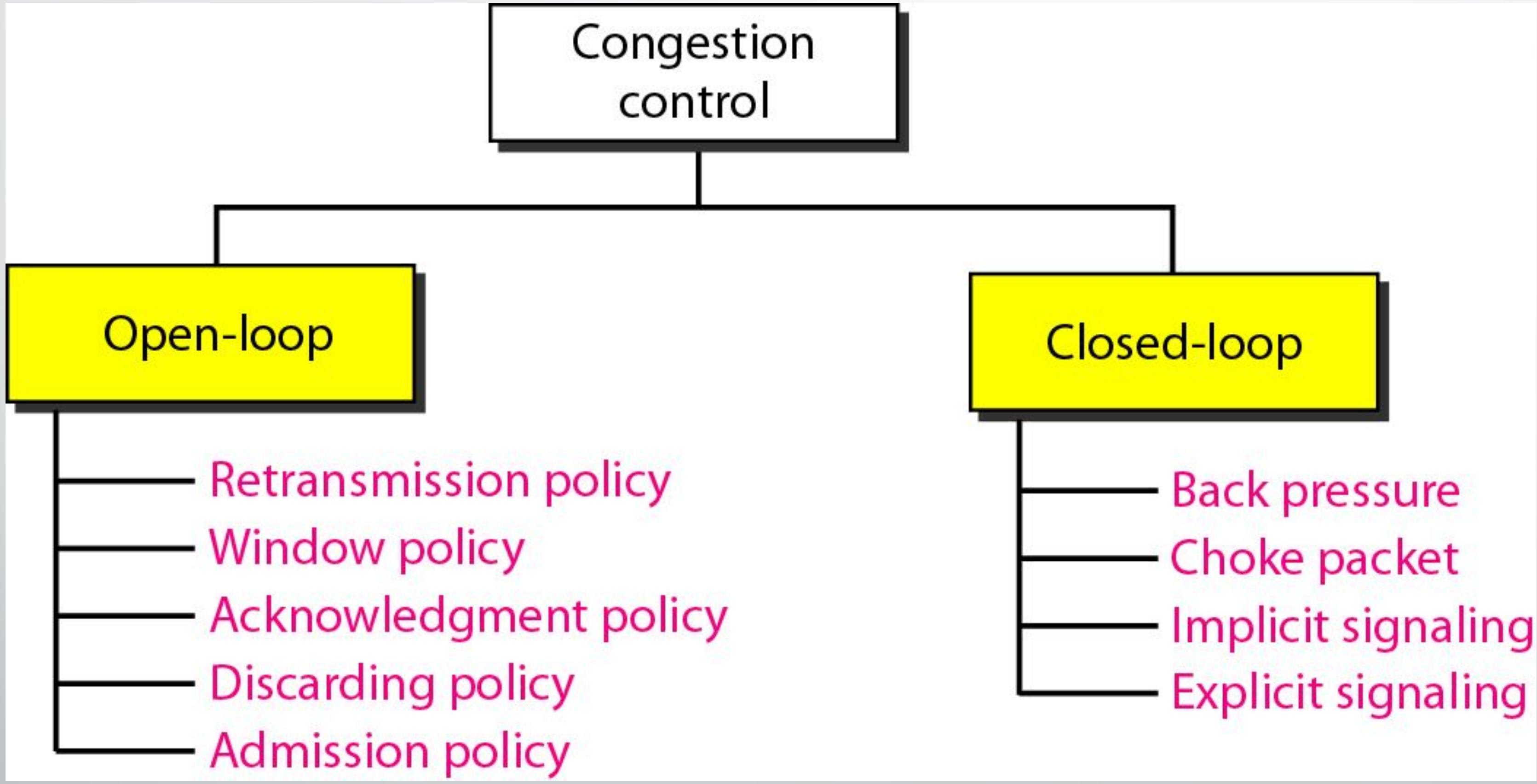
- Why does the throughput sharply decline after the load reaches capacity instead of remaining constant?

Congestion Control

- What is Congestion Control?
  - mechanisms and techniques to control the congestion
  - and keep the load below the capacity.

- Two categories of Congestion Control
  - Open Loop (Prevention)
  - Closed Loop (Removal)

Congestion Control Categories



Open Loop Congestion Control

Open Loop Congestion Control

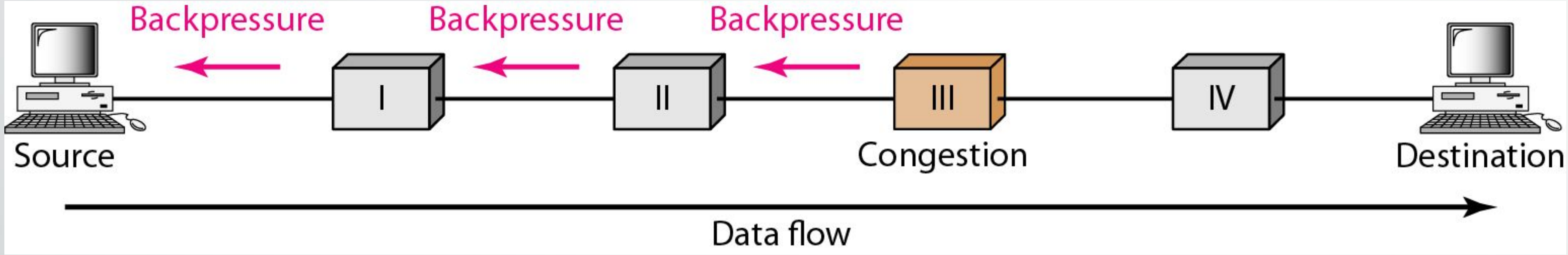
- Retransmission /Window Policy:
  - Retransmission in general increases congestion. (Example-later)
  - Go-Back N ARQ window vs Selective Repeat window.
- Acknowledgement Policy:
  - Not acknowledging every packet slows down sender and helps prevent congestion.
  - Acks are also part of the load in the network.

Open Loop Congestion Control

- Discarding Policy:
  - A good policy by routers may prevent congestion and at the same time may not harm the integrity of the transmission.
- Admission Policy:
  - Check resource requirement before sending packet.
  - Allow no new virtual circuits.

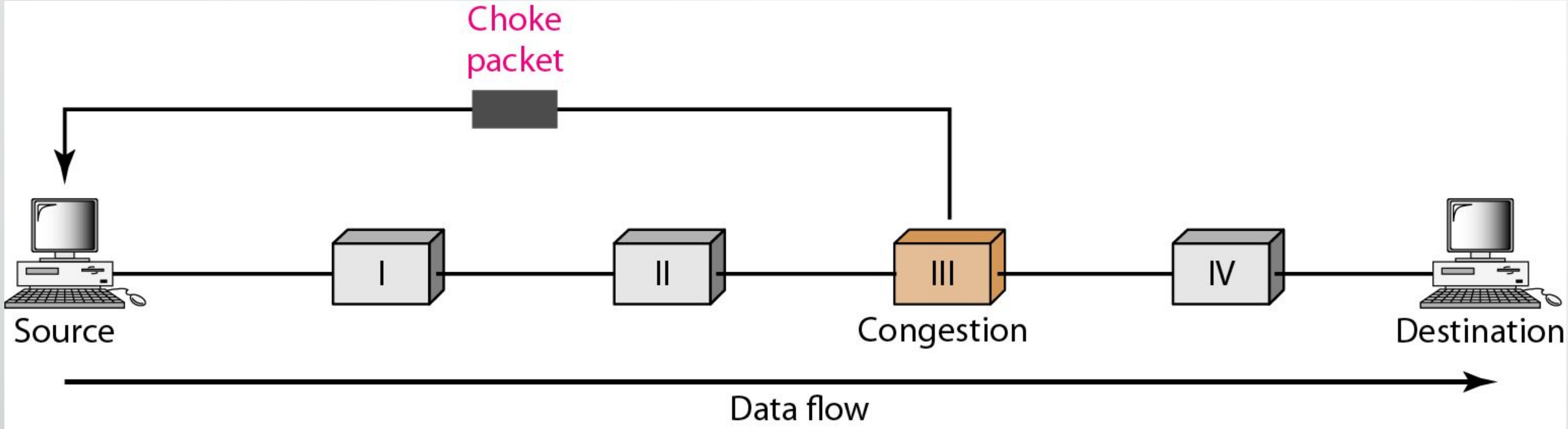
Closed Loop

Backpressure



- Congestion node stops receiving data from upstream nodes.
- Upstream nodes may get congested, they in turn reject data from their upstream nodes.
- Used in Virtual Circuits.

Choke packet



- From a router to source directly.
- Immediate nodes are not warned.
- Example ICMP-source quench message. Immediate routers take no action.

Implicit Signaling

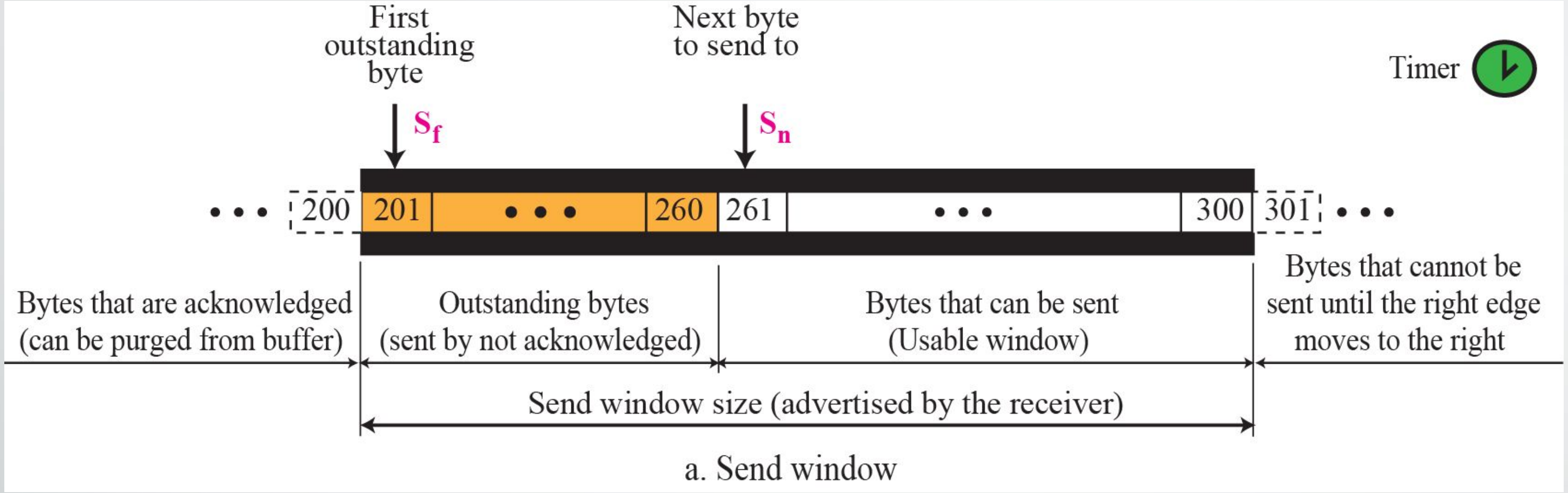
- No communication between the congested node or nodes and the source.
- Source guesses congestion by
  - No acknowledgement for sent packets
  - Delayed acknowledgements
- Then source slows down.

Explicit Signaling

- The node experiencing congestion sends signal to the source.
- Not a separate packet like the “choke”packet.
- Signal included in the data packet itself.
- Can be
  - Backward Signaling-Source warned, slows data
  - Forward Signaling-Receiver warned, slows acks.

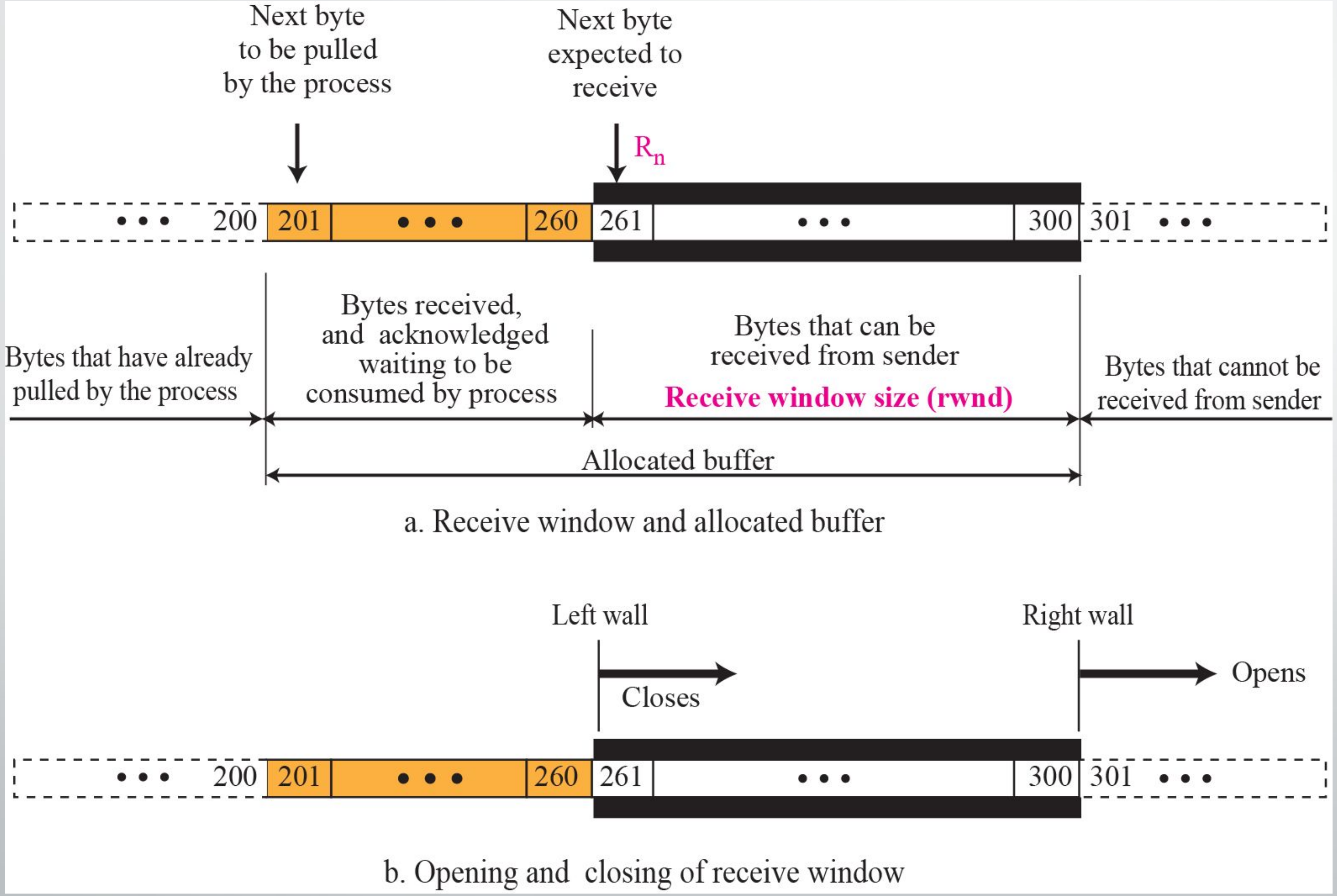
Congestion Control in TCP

TCP Window – Sender Window (Review)

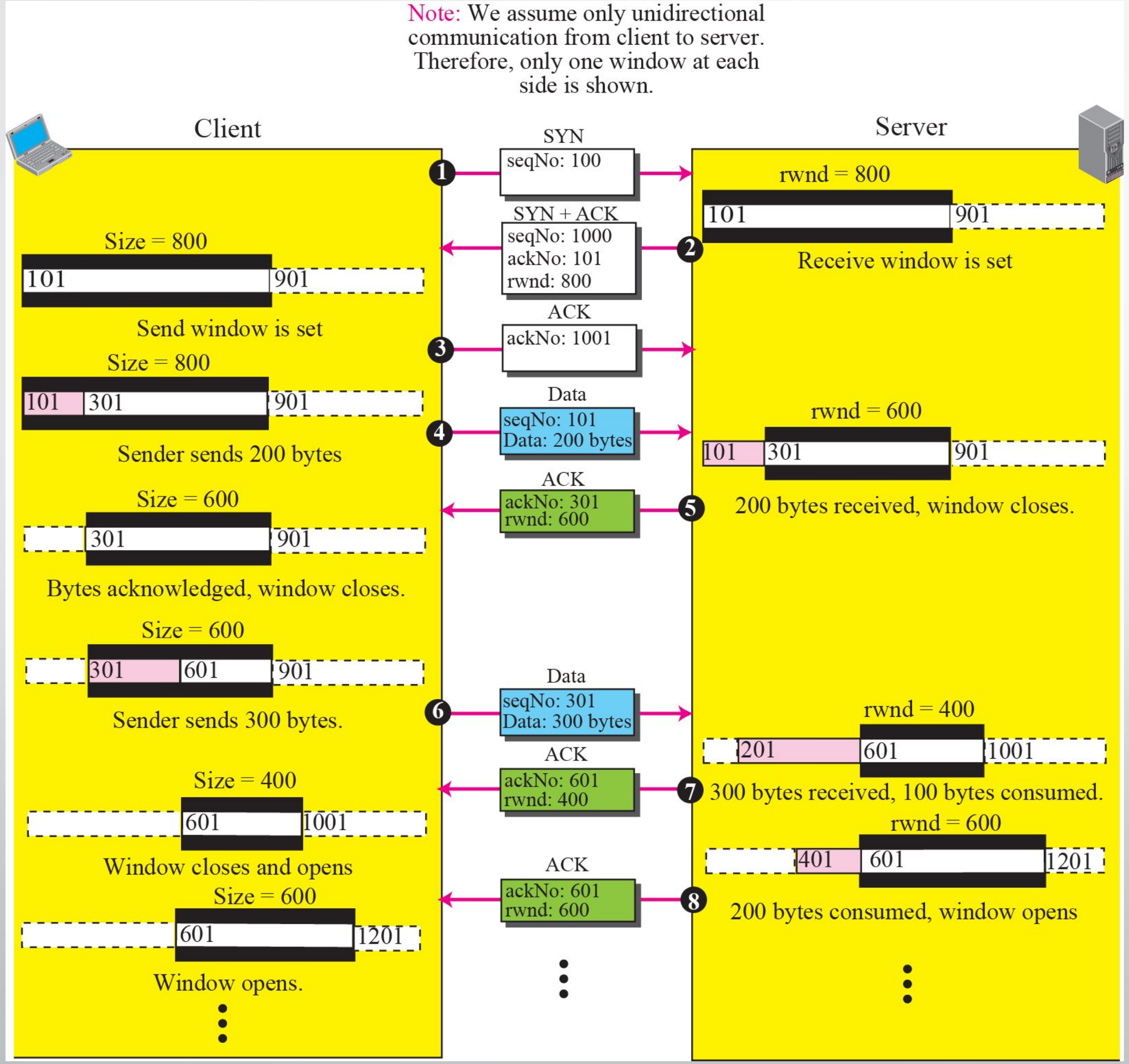


- Sender Window Size is dictated by the receiver window.
- Usually sender window size is determined by the available buffer space in the receiver (rwnd).

TCP Window – Receiver Window (Review)



Normal Flow Control (Review)



TCP Window

- Today, TCP protocols include that the sender's window size is not only determined by the receiver but also by congestion in the network.
- Windows Size of TCP
  - Minimum of rwnd and cwnd
  - Where rwnd is the receiver advertised window size
  - And cwnd is the networks congestion window size

Actual window size = minimum of (rwnd,cwnd)

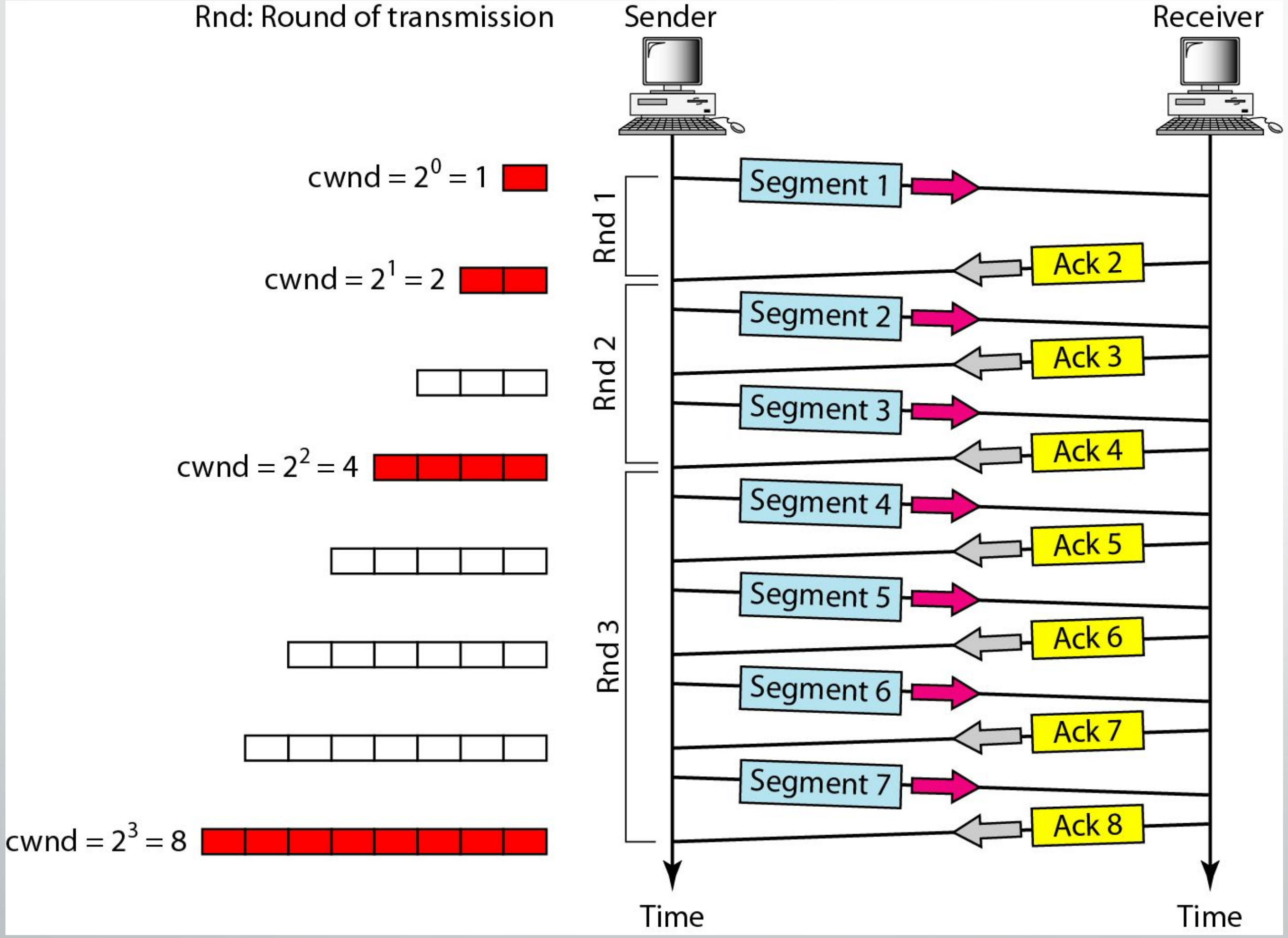
TCP congestion control

- TCP does congestion control in three phases:
  - Slow Start
  - Congestion Avoidance
  - Congestion Detection

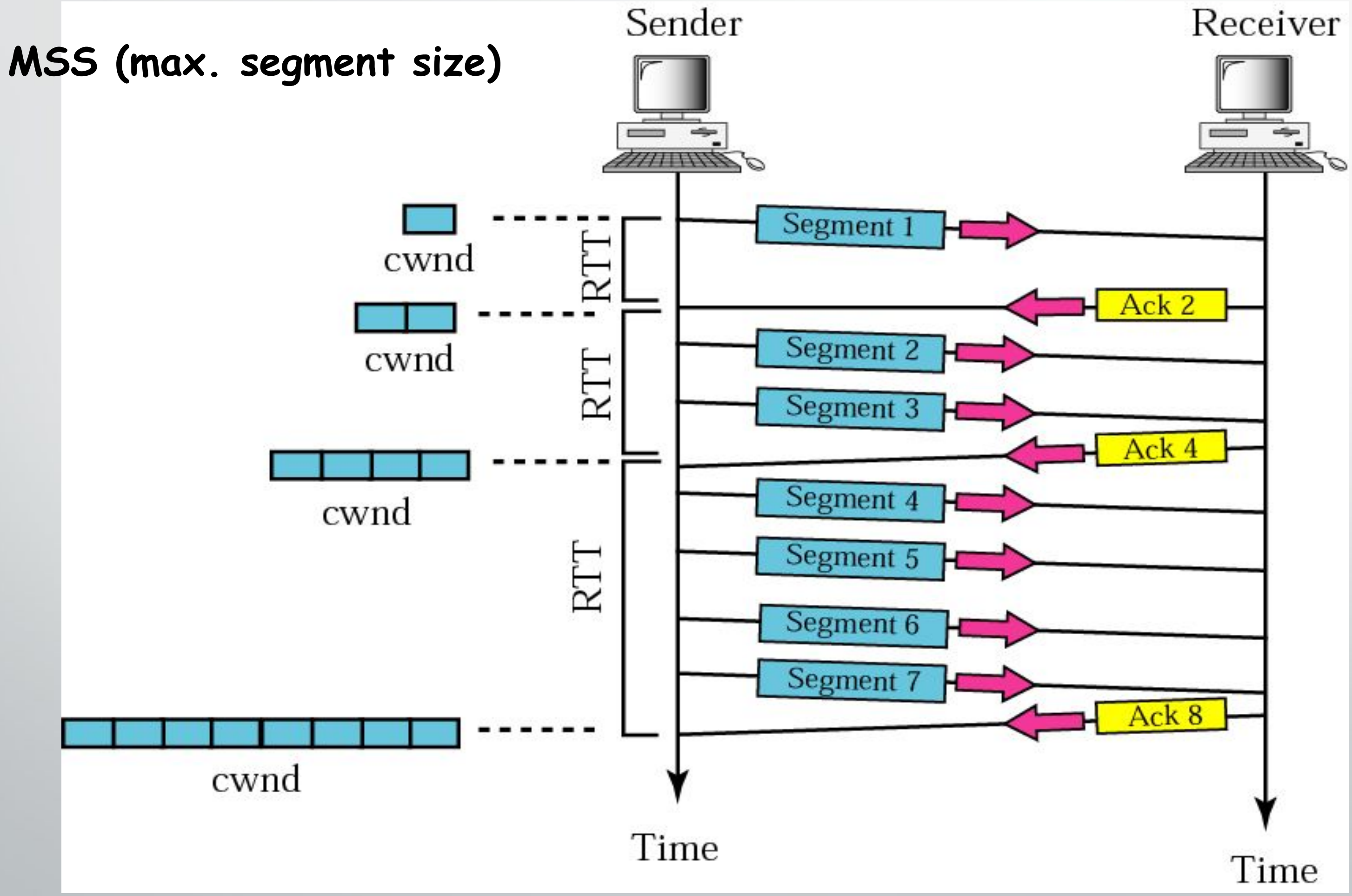
Slow Start

- cwnd starts with one maximum segment size(MSS).
- MSS determined during connection establishment.
- MSS increases exponentially after each acknowledgement.

Slow Start- Exponential Increase



Slow Start- Exponential Increase

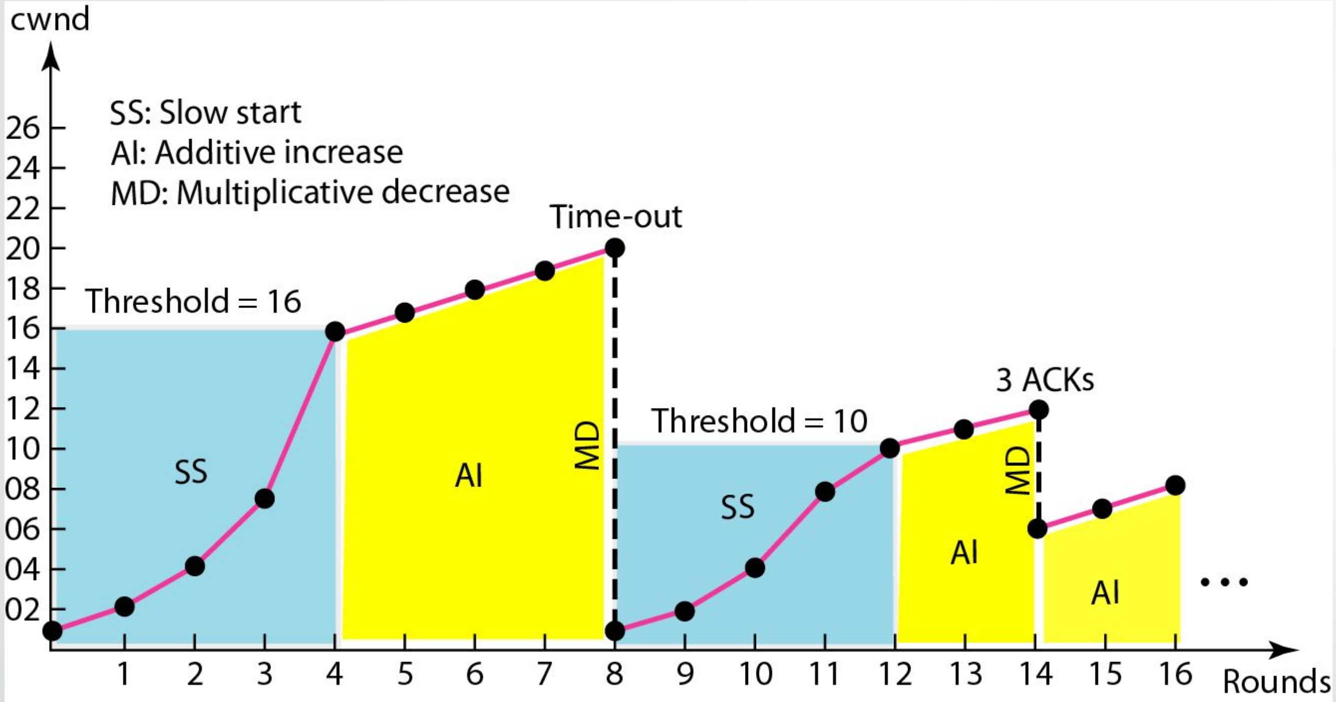


Slow Start- Exponential Increase

- Assumptions:
- rwnd>cwnd , so sender window=cwnd
- Each segment 1 byte
- Each segment is acknowledged individually*.

In the slow-start algorithm, the size of the congestion window increases exponentially until it reaches a threshold.

Slow Start

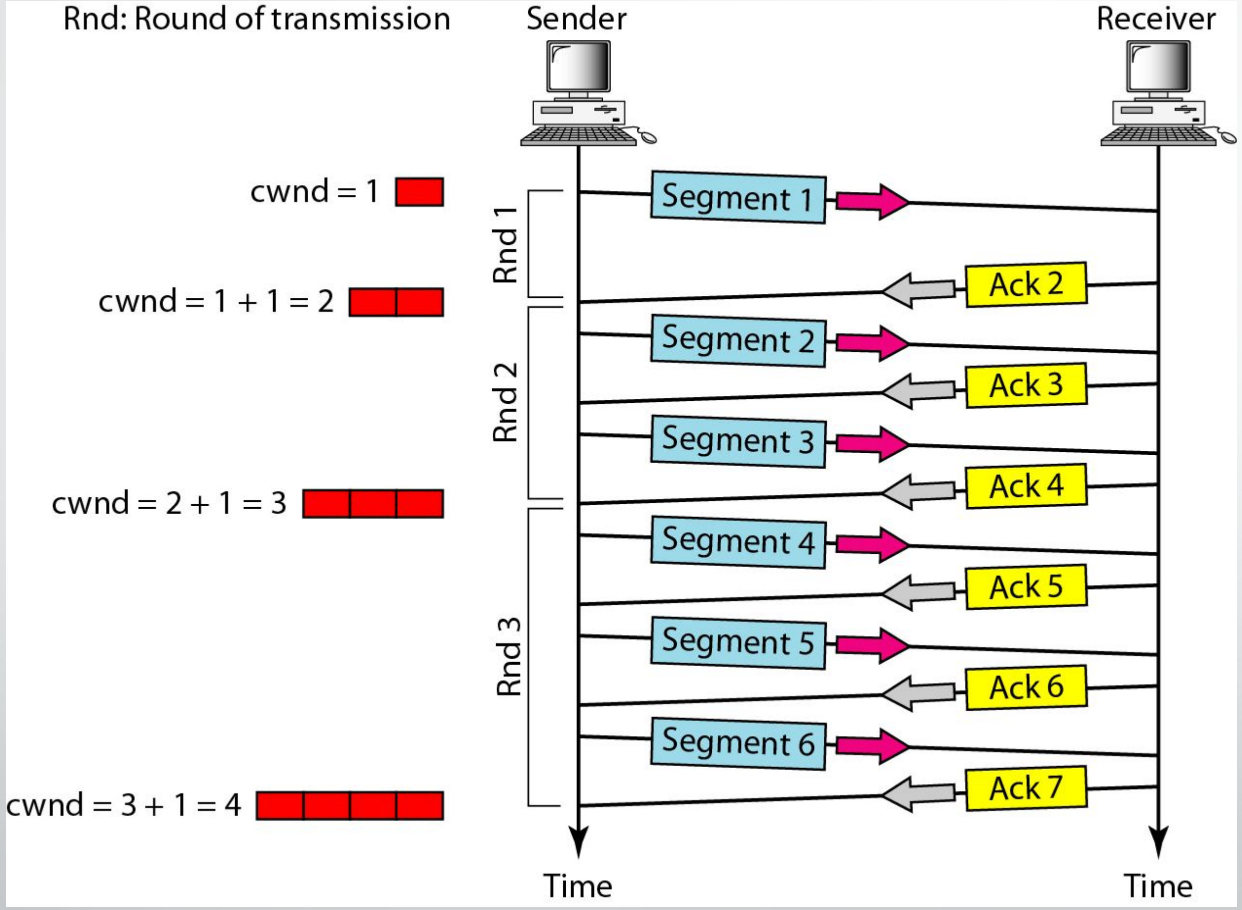


Sender keeps track of a variable named *ssthresh.* When window reaches *ssthresh* the next phase starts.
Most implementation *ssthresh* is 65,535 bytes.

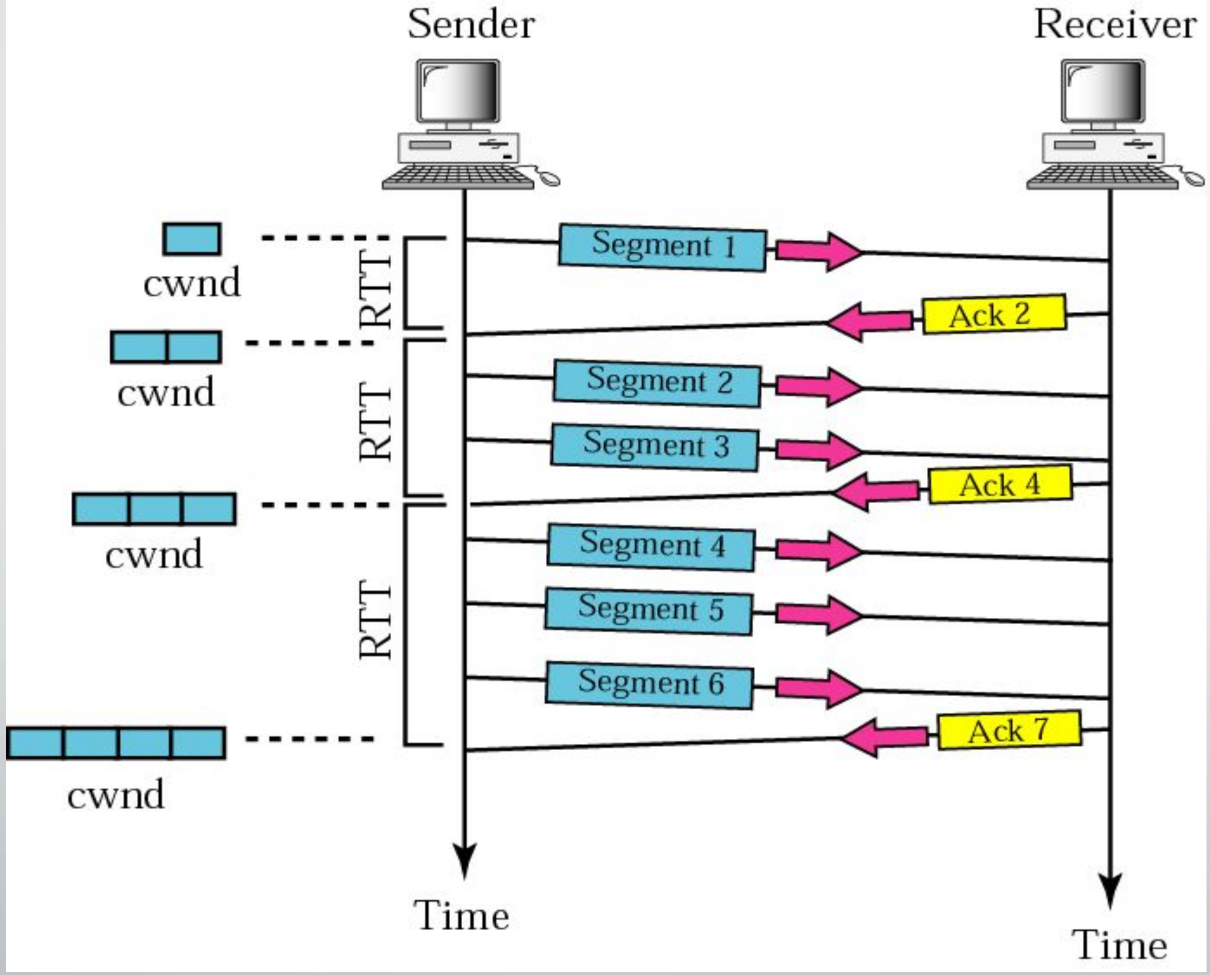
Congestion Avoidance:

In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.

Congestion Avoidance:



Congestion Avoidance:



Congestion Detection:

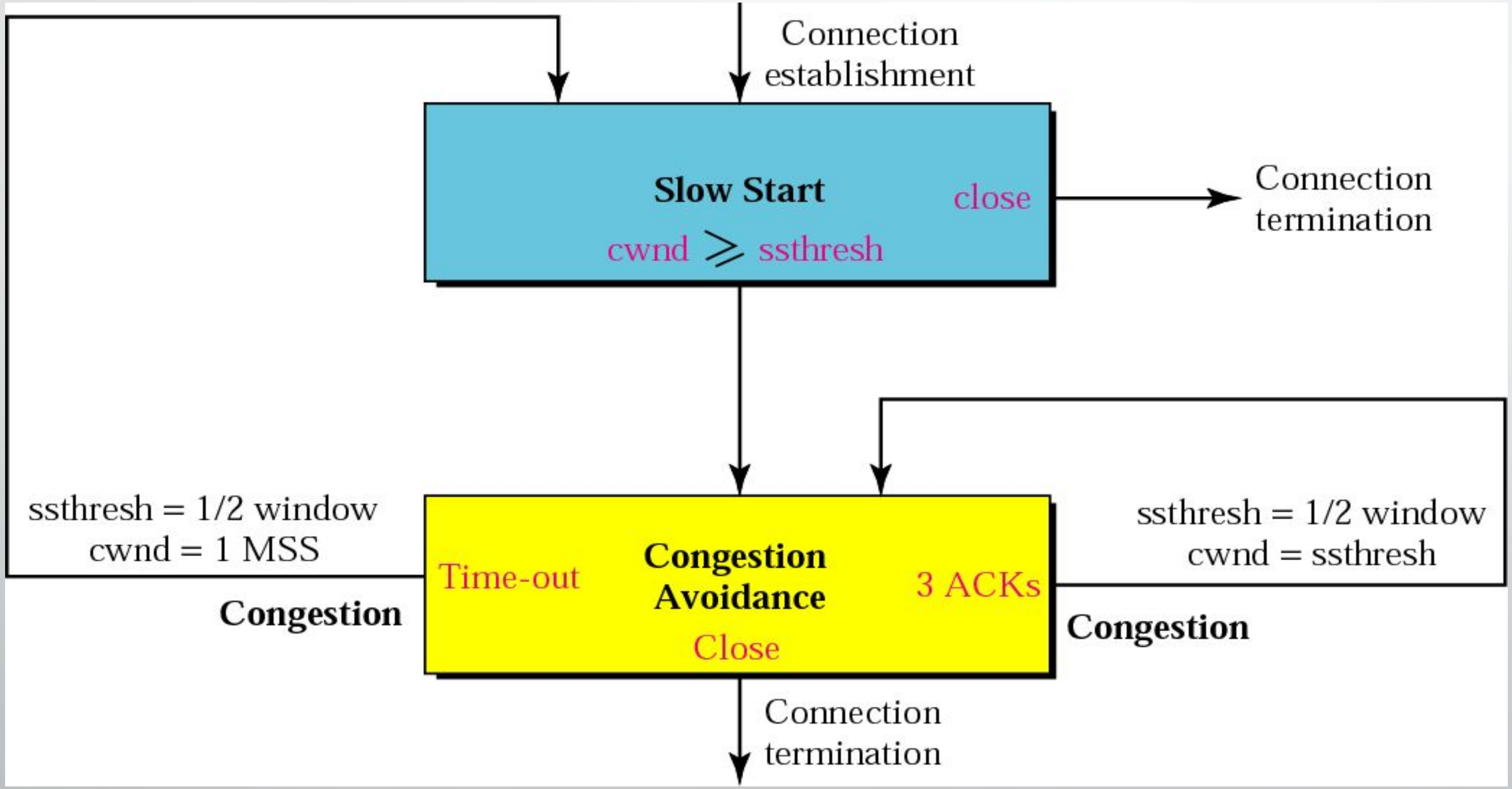
An implementation reacts to congestion detection in one of the following ways:

- If detection is by time-out, a new slow start phase starts.
- If detection is by three ACKs, a new congestion avoidance phase starts.

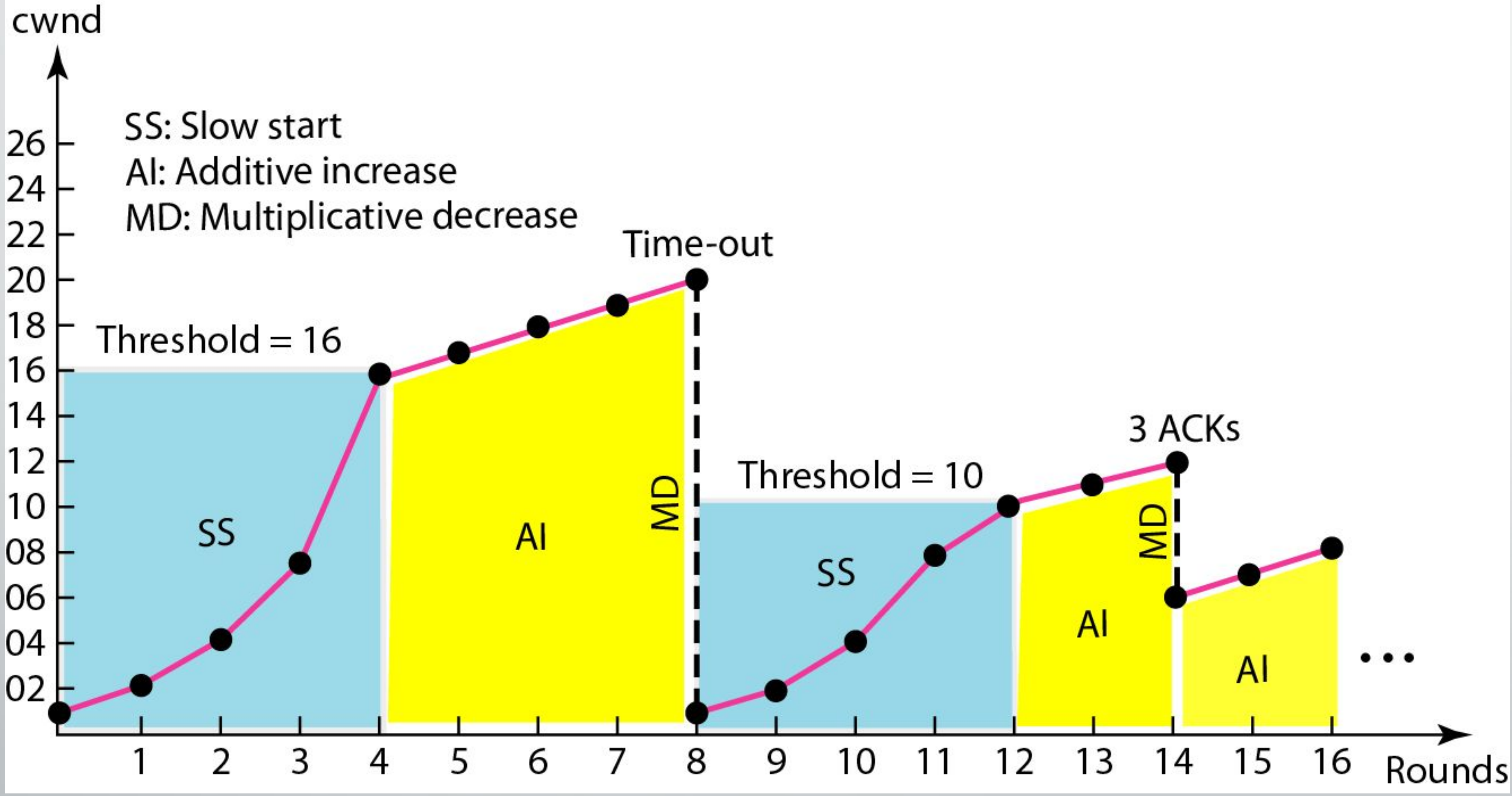
Congestion Detection:

- Congestion detected by time-out (severe congestion)
  - starts new slow start phase
  - ssthresh = max(2, floor(cwnd/2))
  - cwnd=1
- Congestion detected by three ACKs (less severe congestion)
  - starts new congestion avoidance phase
  - ssthresh = max(2, floor(cwnd/2))
  - cwnd= ssthresh

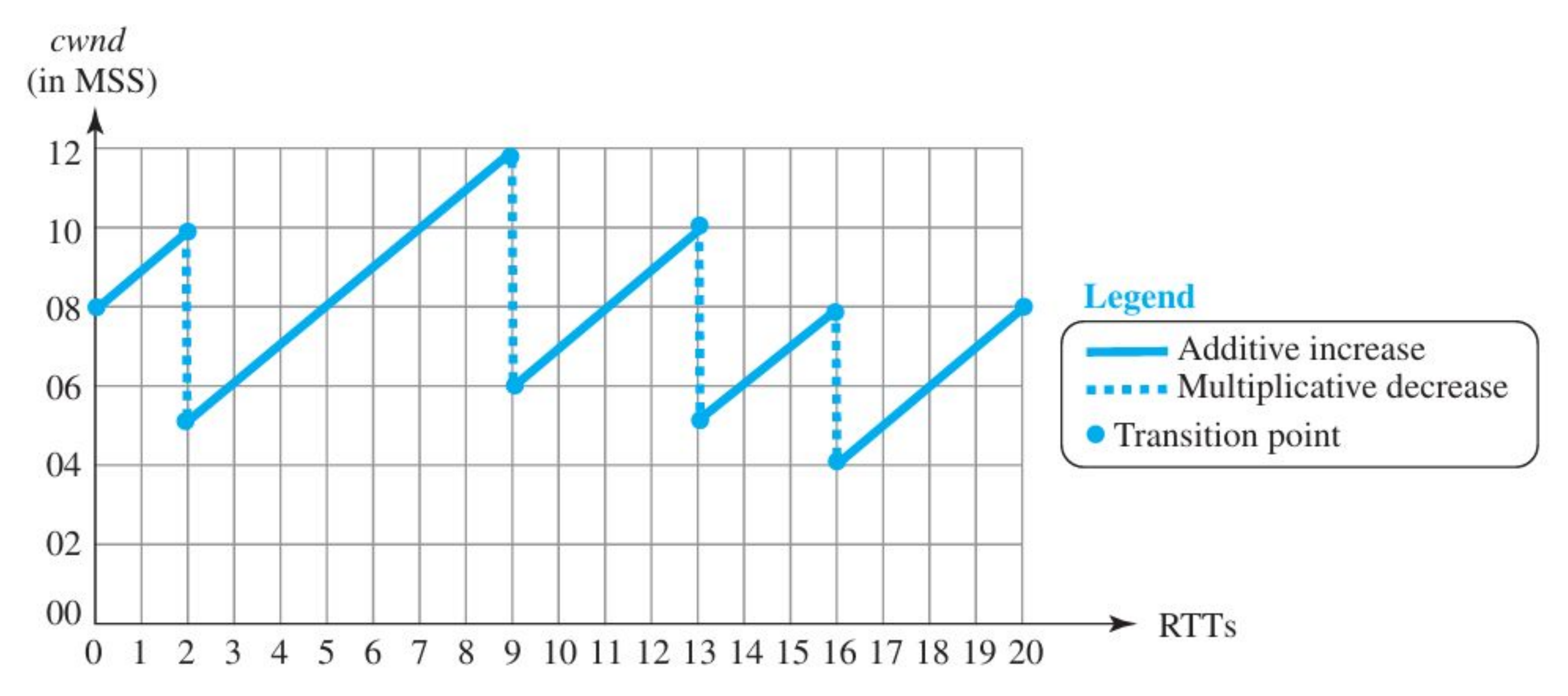
TCP Congestion Policy:



TCP Congestion Example:



AIMD: Additive Increase Multiplicative Decrease



The End