Week 8: Transport Layer Contd

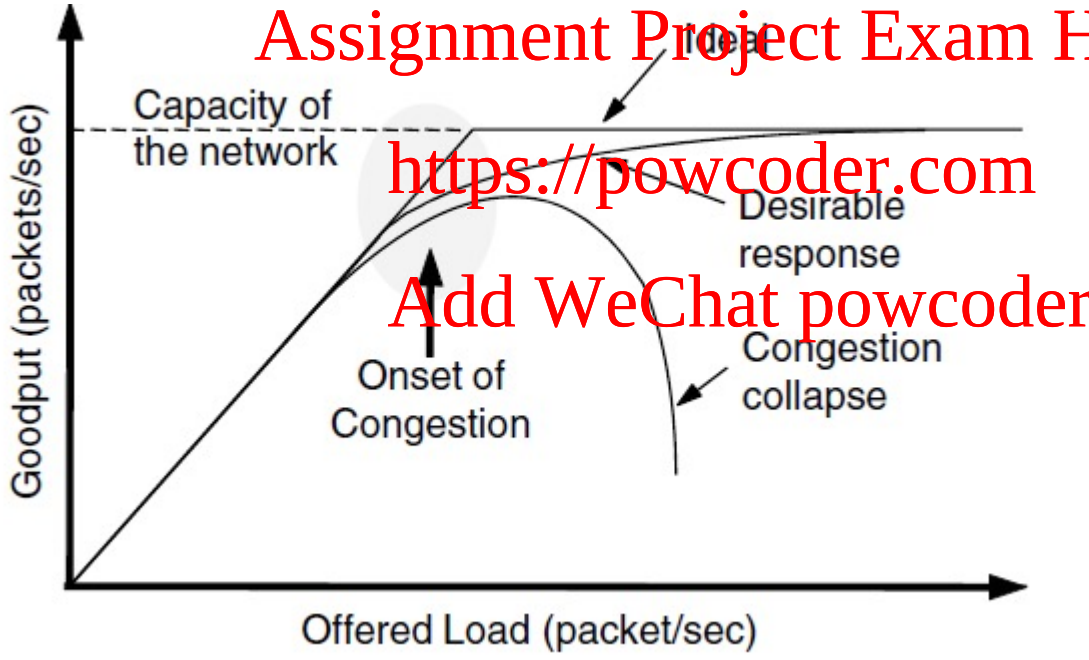
Internet Technologies COMP90007

Lecturer: Tom Drummond

Semester 2, 2021

What happens when congested?

- Congestion results when too much traffic is offered; performance degrades due to loss/retransmissions
- Goodput (=useful packets) trials offered load



Congestion Control vs Flow Control

- Flow control is an issue for point to point traffic, primarily concerned with preventing sender transmitting data faster than receiver can receive it
- Congestion control is an issue affecting the ability of the subnet to actually carry the available traffic, in a global context

Load Shedding

- When congestion control mechanisms fail, load shedding is the key remaining possibility - **drop packets**
- In order to ameliorate impact, applications can mark certain **packets as priority** to avoid discard policy

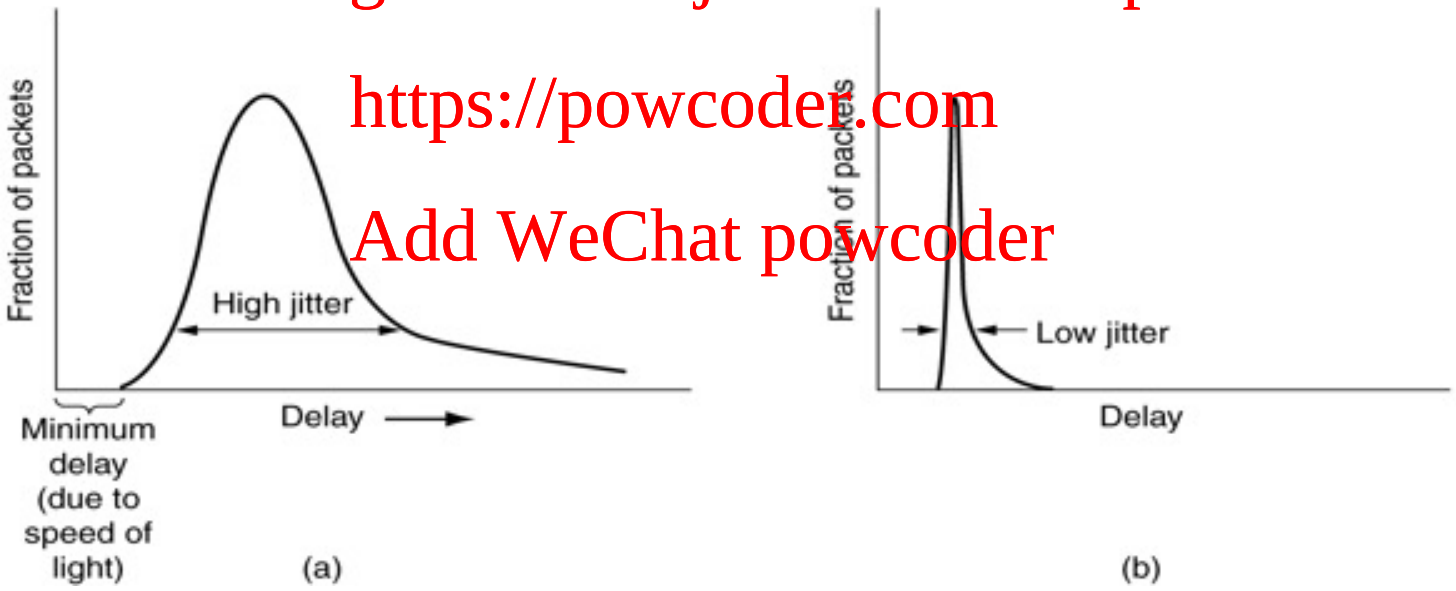
What is the key problem if network is not delivering properly:

- Quality of Service becomes low
- **Expected network performance** is an important criterion for a wide range of network applications
- Some **engineering techniques are available to guarantee QoS** (Quality of Service)
- 4 things to watch out for:

  bandwidth, reliability, delay, jitter

Jitter is Interesting/New for Us

- Jitter is the **variation in packet arrival times**
  - a) high jitter
  - b) low jitter



Mechanisms for Jitter Control

- Jitter is an issue for some applications
- Jitter can be contained by **determining the expected transit time** of a packet
- Packets can be **shuffled at each hop in order to minimise jitter** - slower packets sent first, faster packets wait in a queue
- For certain applications jitter control is extremely important as it mainly directly affects the **quality perceived by the application user**

QoS Requirements

- Different applications care about different properties
  - We want all applications to get what they need

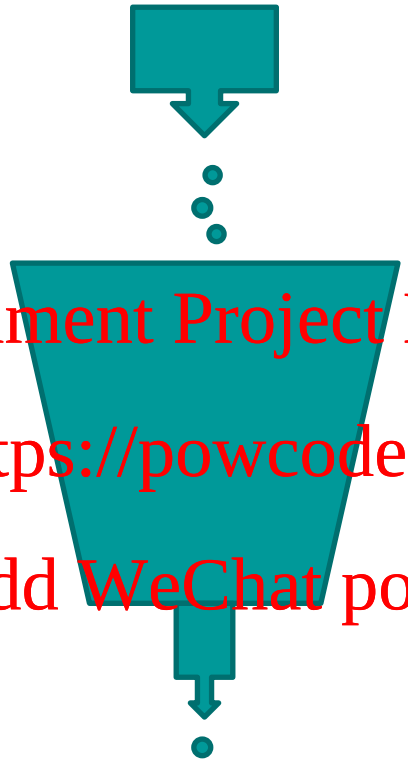
"High" means demanding the requirement!

Application	Bandwidth	Delay	Jitter	Loss
Email	Low	Low	Low	Medium
File sharing	High	Low	Low	Medium
Web access	Medium	Medium	Low	Medium
Remote login	Low	Medium	Medium	Medium
Audio on demand	Low	Low	High	Low
Video on demand	High	Low	High	Low
Telephony	Low	High	High	Low
Videoconferencing	High	High	High	Low

Techniques for Achieving QoS

- **Over-provisioning**
  - more than adequate buffer, router CPU, and bandwidth (expensive and not scalable ...)
- **Buffering**
  - buffer received flows before delivery - increases delay, but smoothes out jitter, no effect in reliability or bandwidth
- **Traffic Shaping**
  - regulate the average rate of transmission and burstiness of transmission
  - **leaky bucket**
  - **token bucket**

Leaky Bucket



Large **bursts** of traffic is buffered and smoothed while sending

e.g. can be done at host sending data

Techniques for Good QoS Contd

- **Resource reservation**
  - reserve bandwidth, buffer space, CPU in advance
- **Admission control**
  - routers can decide based on traffic patterns whether to accept new flows, or reject them
- **Proportional routing**
  - traffic for same destination split across multiple routes
- **Packet scheduling**
  - Create queue(s) based on priority etc
  - fair queuing, weighted fair queueing

TCP and Congestion Control

- When networks are overloaded, congestion occurs, potentially affecting all layers
- Although lower layers (data and network) attempt to ameliorate congestion, in reality **TCP impacts congestion most significantly** because TCP offers best methods to reduce the data rate, and hence reduce congestion itself

Congestion Control: Design

- Two different problems exist
  - network capacity and receiver capacity
  - these should be dealt with separately, but compatibly
- The sender maintains two windows actually
  - Window described by the receiver
  - Congestion window
- Each regulates the number of bytes the sender can transmit – the maximum transmission rate is the **minimum of the two windows**

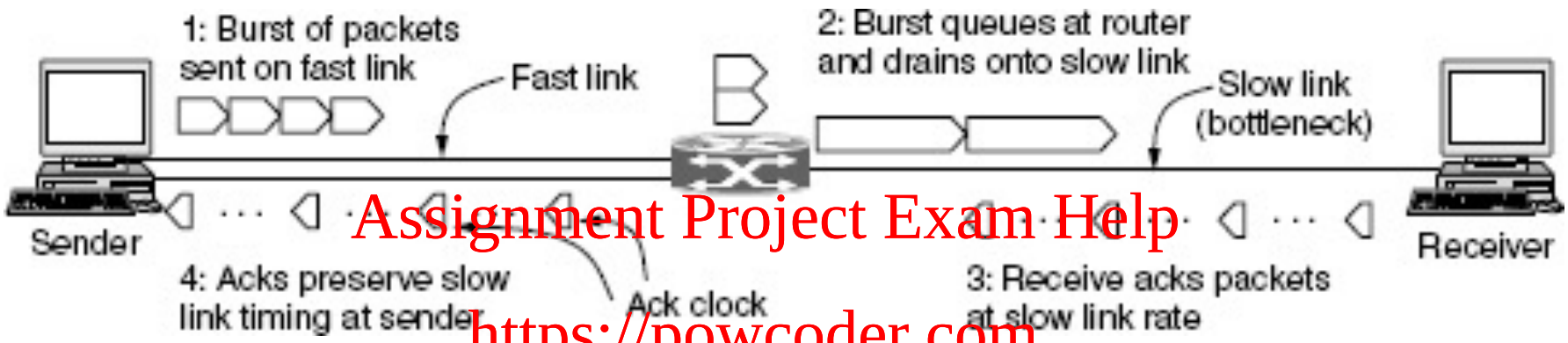
TCP and Congestion Control Contd

- TCP adopts a defensive stance:
  - At connection establishment, a **suitable window size is chosen by the receiver based on its buffer** size
  - If the sender is constrained to this size, then **congestion problems will not occur due to buffer overflow** at the receiver itself, but may still **occur due to congestion within the network**

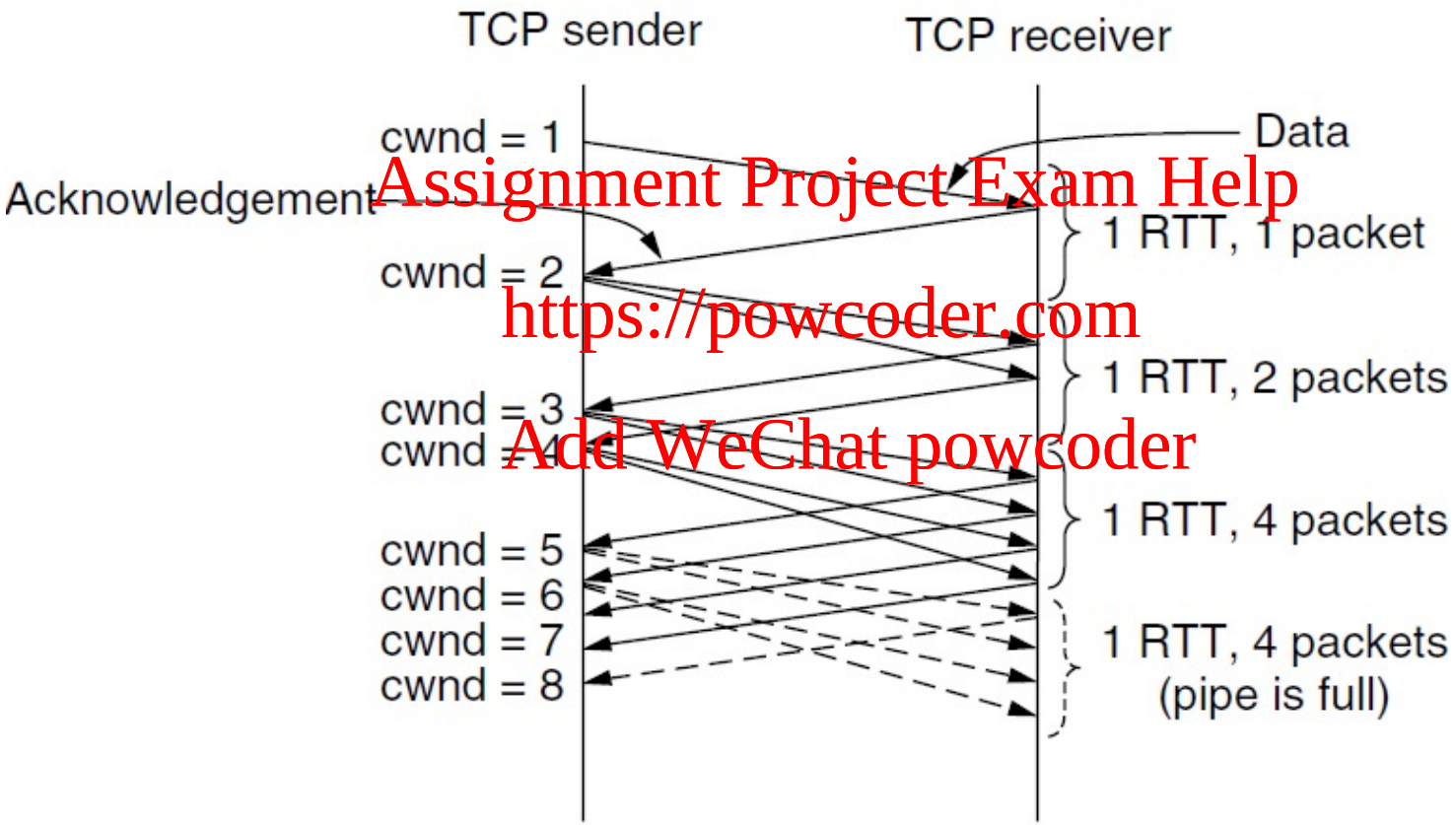
Incremental Congestion Control: Slow Start

- On connection establishment, the **sender initializes the congestion window to a size**, and transmits one segment
- If this segment is acknowledged before the timer expires, **the sender adds another segment's worth of bytes to the congestion window**, and transmits two segments
- As **each new segment is acknowledged**, the congestion window is increased by **one more segment**
- In effect, each set of acknowledgements doubles the congestion window - which **grows until either a timeout occurs or the receiver's specified window is reached**

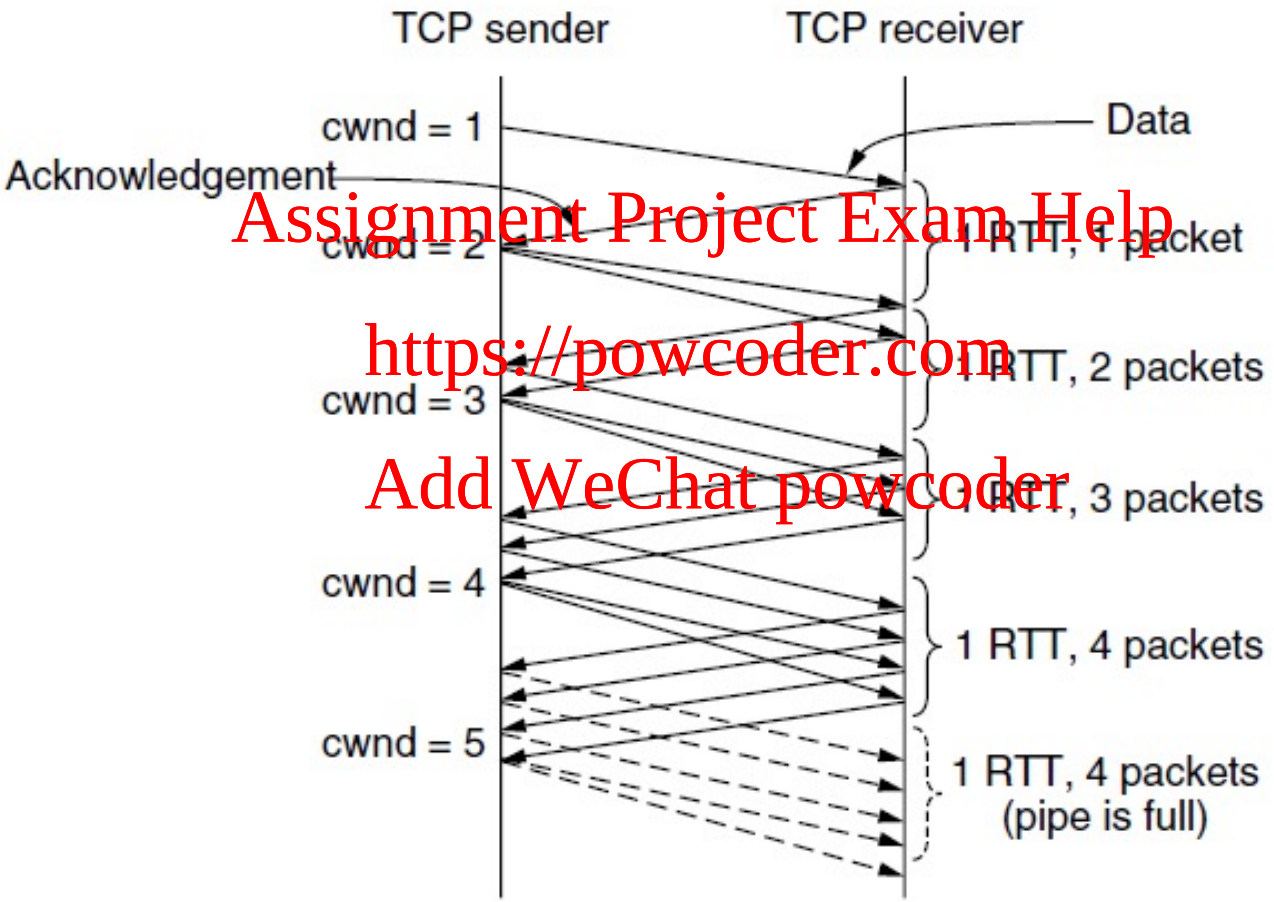
Ack Clock



Slow Start (badly named…)



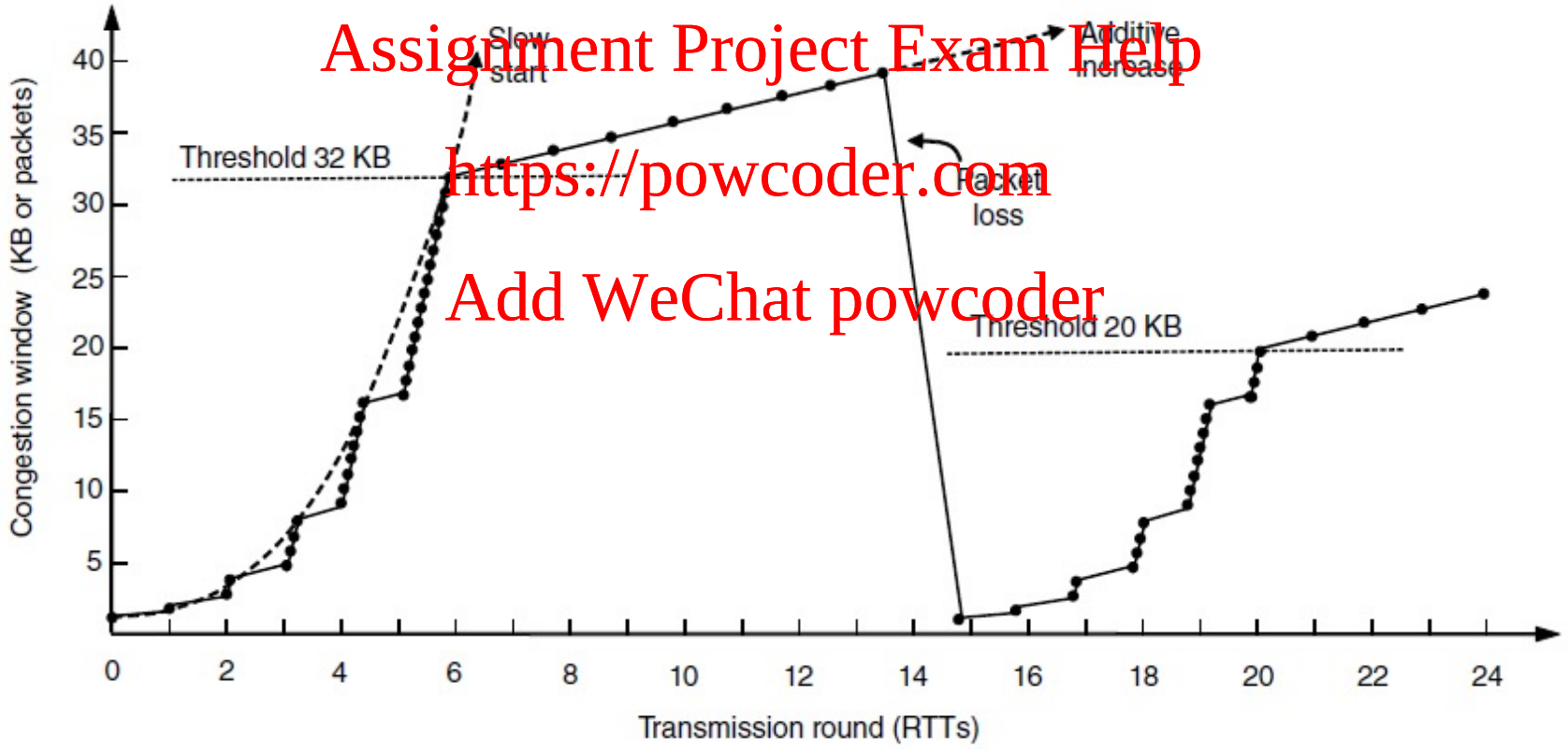
Additive increase



Internet Congestion Control

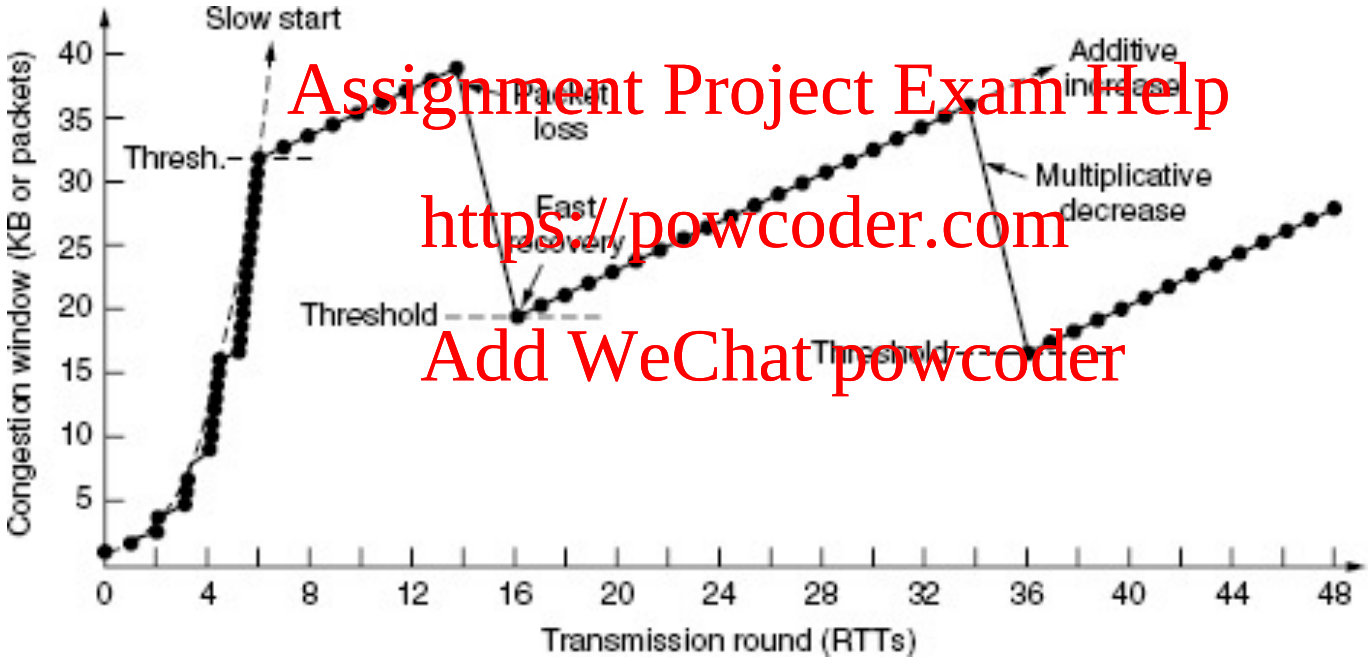
Slow start followed by additive increase (TCP Tahoe)

Threshold is half of previous



Internet Congestion Control Contd

Another one with TCP Reno



Congestion Control And Wireless

- Much harder to deal with
  - Things are increasingly wireless
  - Not everything is wireless, but parts of a path
  - So how does one know where wireless is
  - More variety on wireless links as well
  - SNR varies when people move
  - Delay is different if it is Wifi vs Satellite
  - This is a hot area of research…

TCP Timer

- A key worry is when timers go out
- Too early means too many resends
- Too late means reliability comes with more additional cost
- Solutions rely on dynamicity as network conditions change
- One needs to measure network performance and adapt timers