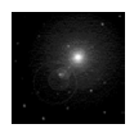
TCP Congestion Control

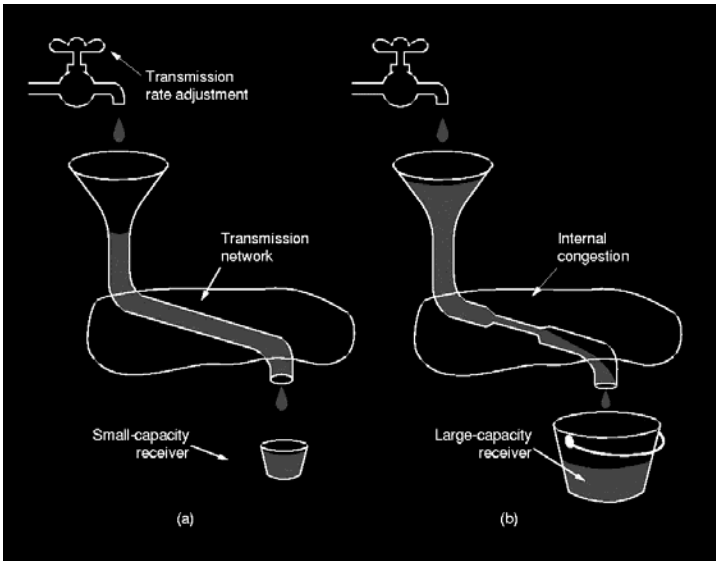
TCP Congestion Control



COMPUTER
COMMUNICATION
NETWORK

- Congestion management involves both Network and the End Stations.
- Ultimately TCP has do bulk of the work. Why?
- How to sense congestion?

Two Sources of Congestion



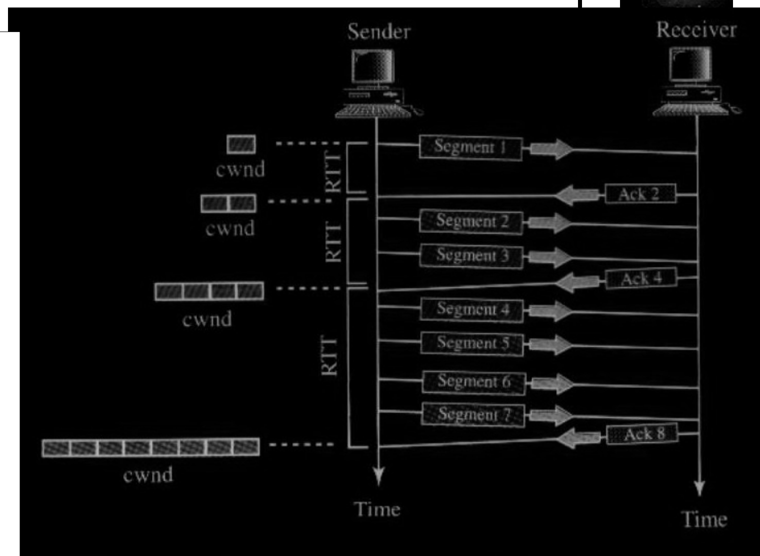
COMPUTER
COMMUNICATION
NETWORK

TCP Solution

- TCP Maintains two windows:
  - CON.WINDOW to track network limits.
  - ADV.WINDOW to track receiver limits.
- MAXWINDOW
  - MIN(CON.WINDOW, ADV.WINDOW)
- EFFECTIVEWINDOW
  - MAXWINDOW - (LASTBYTESENT-LASTBYTEACKED)

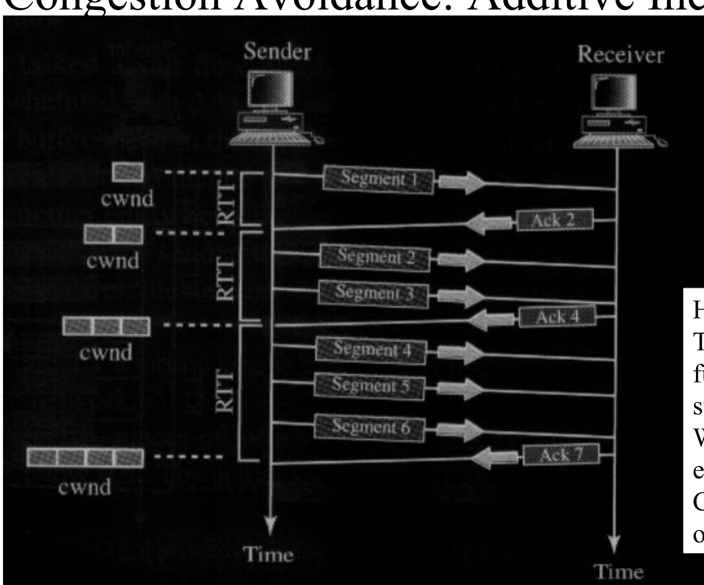
COMPUTER
COMMUNICATION
NETWORK

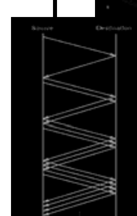
Slow Start!



- Initially TCP sets CON.WINDOW=MTU.
- It first sends 1 packet with MTU bytes.
- When it receives 1st ACK, it doubles CON.WINDOW and sends out 2 packets in a burst.
- When it receives the 2nd ACK, it then sends out 4 packets in a burst...
- CON.WINDOW keeps growing until either a timeout occurs or receiver's window is reached.

Congestion Avoidance: Additive Increase

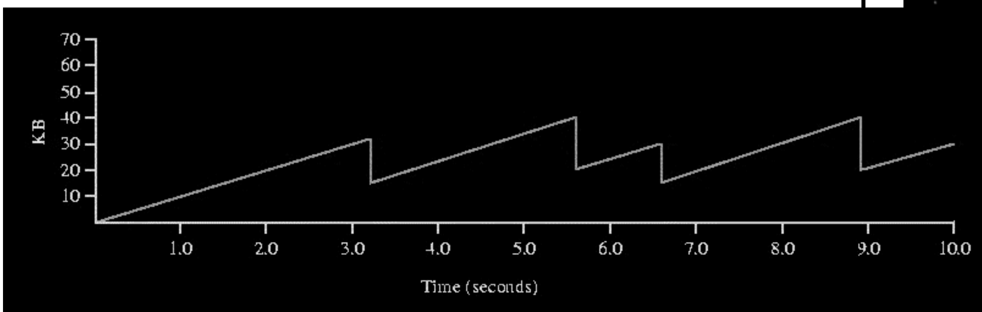




PUTER
NICATION
VORK

However, a third parameter THRESHOLD is also used to further slow down the slow start process.
When the CON.WINDOW exceeds threshold, the CON.WINDOW is increased only once for a burst...

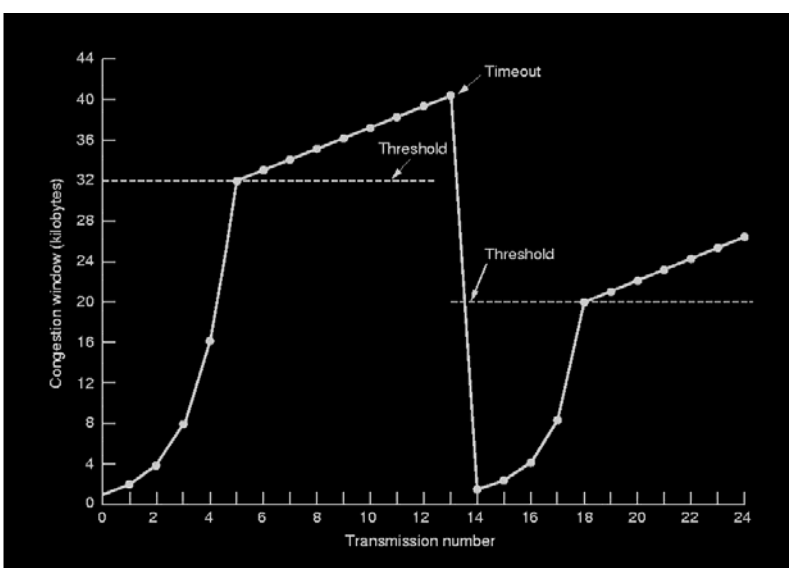
Congestion Detection: Multiplicative Back-off



ER
ATION
K

If, there is a timeout the CON.WINDOW is reduced to half. If again the timeout occurs it is reduced to half again. Until it becomes 1 MTU.
It cannot go down below 1 MTU bytes.
Each time a timeout occurs, THRESHOLD is set to half of current CON.WINDOW

TCP Congestion Windowing

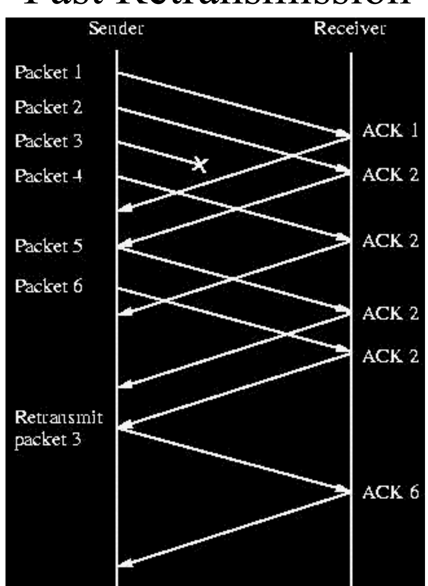


COMPUTER
COMMUNICATION
NETWORK

Fast Retransmit and Fast Recovery

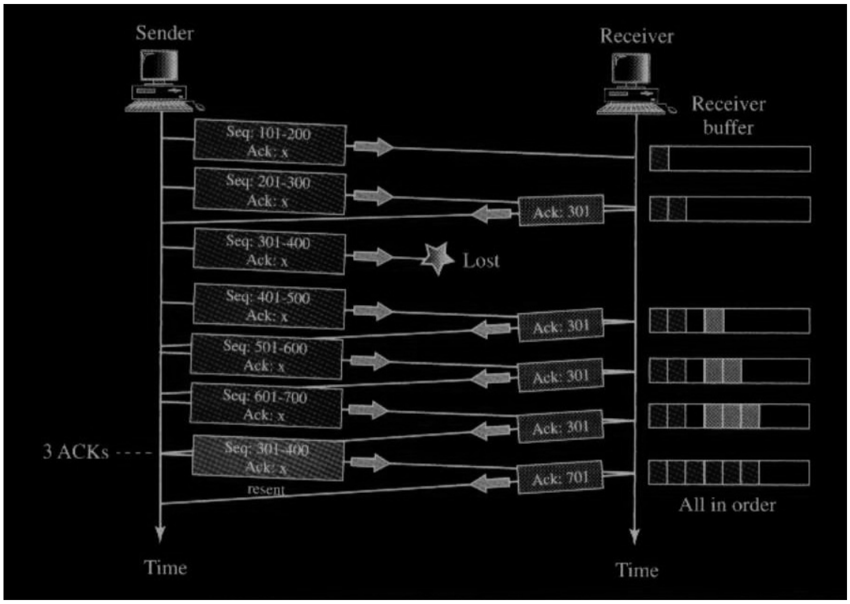
- If a packet is lost TCP can sense it by a mechanism of multiple repeat acknowledgements, even before the timer goes out!
- Even if it receives a packet out of order. It sends out an ACK, for the last byte received in order.
- By sensing repeated ACK for many times (more than 3 generally), the sender can figure out probably the packet in between is lost.
- It immediately retransmits the suspected packet, even if the timer is not out.

Fast Retransmission



COMPUTER
COMMUNICATION
NETWORK

Fast Retransmission



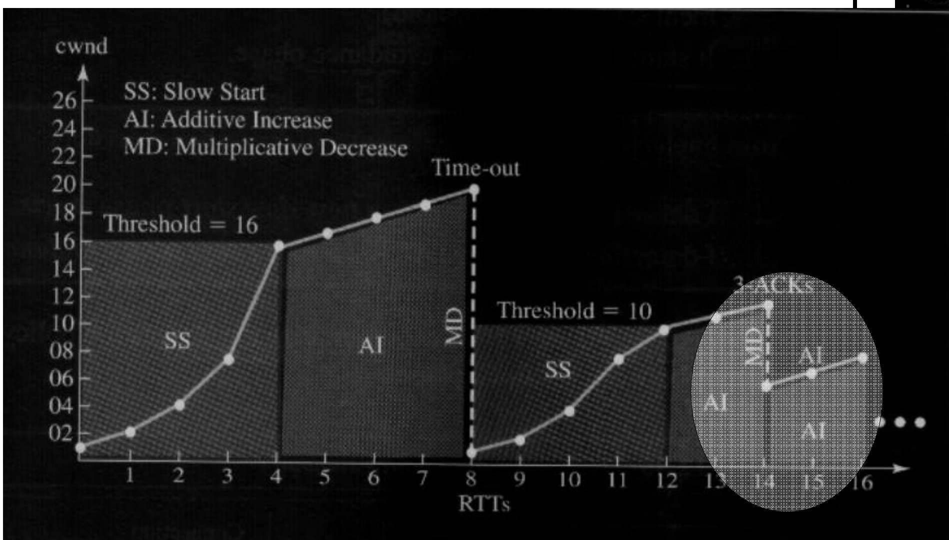
COMPUTER COMMUNICATION NETWORK

Why Fast Retransmit?

- Is it congestion or a packet loss?
- Since ACKs are coming there is less chance of a congestion.
- Thus it is managed as following:
  - Half's the CON.WINDOW and THRESHOLD (it is still cautious!)
  - And moves directly into additive increase mode. Jumps to WINDOW=THRESHOLD rather than from WINDOW=1

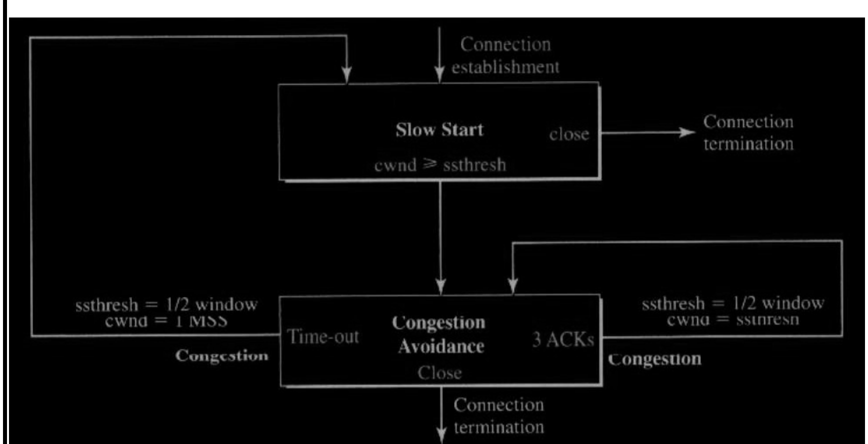
COMPUTER COMMUNICATION NETWORK

Windowing with Fast Retransmit



UTER
CATION
ORK

TCP Congestion Policy



COMPUTER
COMMUNICATION
NETWORK

TCP Congestion Avoidance Mechanisms

DECbit Mechanism

- The idea is to avoid congestion altogether.
- DECbit algorithm:
  - Routers sends a special bit if it senses growing queue length.
  - The receiver, upon receiving the packets, sets the bit in the acknowledgement too.
  - Router measures the average queue length over current load cycle+last free cycle + last load cycle.
  - If it is larger that 1, it sets the bit.
  - If host records how many of its packets got the bit.
  - If less than 50% it increases the CON.WINDOW by 1
  - If it is greater, it decreases the CON.WINDOW by .875

That is the additive increase/ multiplicative decrease rule.

Random Early Detection (RED) Gateway

- No bits are set. There is no explicit message.
- But, routers intentionally drops a packet to send an implicit message to the sender, even if it is not fully congested, but when the congestion is building up.
- The receiver get notified by the timeout.
- Its timeout mechanism reduces the flow rate.

Source-based Congestion Avoidance (TCP Vegas)

- No role of the network elements.
- The source cautiously observers for various signs of congestion. Such as:
  - RTT is increasing measurable for each successive packet transmission.
  - The sending rate or throughput is flattening.

Conclusion

- Congestion control is one of the most actively researched area in networking.
- With the arrival of newer technologies (wireless, high speed fiber) it is getting more complex.
- An open problem and place to make your mark!

COMPUTER
COMMUNICATION
NETWORK

Virtual Clock

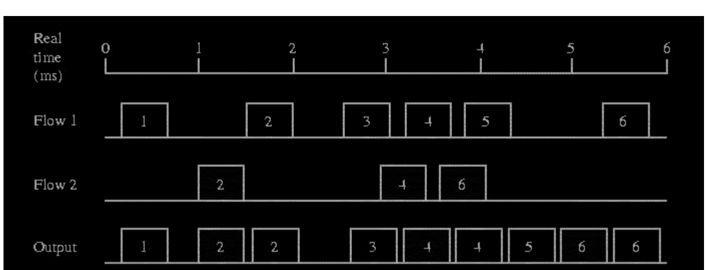
Flows Using Virtual Clock

- It is a reservation based flow control.
- Each flow request connection with two parameters *average rate* (AR) and average interval (AI).
- Each flow gets a virtual clock.
- Each time a packet arrives it generates a some ticks
  - Virtual Ticks (i)= 1/AR(i)
- The virtual clock is advanced
  - Virtual Clock += Virtual Tick(i)
- Packets get a time stamp and are added to the common queue sorted by their timestamp.
- If congestion occurs, packet with highest time stamp is dropped.

COMPUTER COMMUNICATION NETWORK

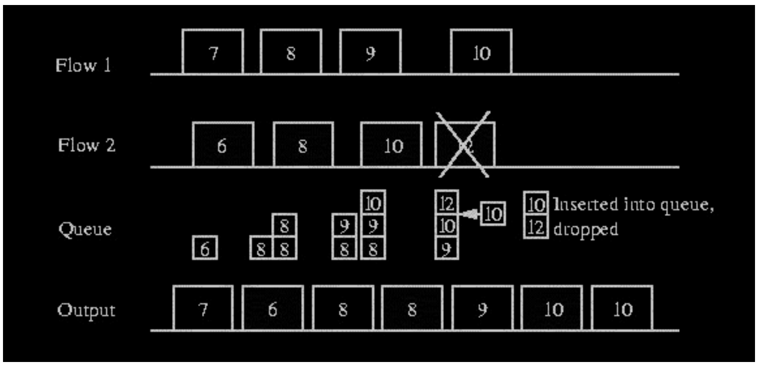
Example of Virtual Clock Flow Control

- Flow 1 has TA= 1000 packets/sec
  - Virtual Tick = 1 ms.
- Flow 2 has TA= 500 packets/sec
  - Virtual Tick = 2 ms.



COMPUTER COMMUNICATION NETWORK

Packet Discarding using VC Algorithm



In this scheme, if nobody is sending, one host can steal cycles. But, if there is congestion, one cannot disrupt others rate.

COMPUTER COMMUNICATION NETWORK

What if one does not send for a long time? Its Virtual Clock does not increase. Then it can suddenly send a big burst disrupting others. Solution?