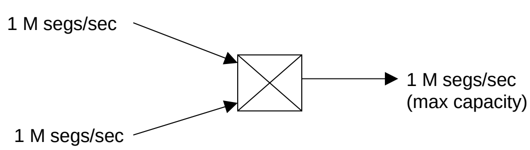
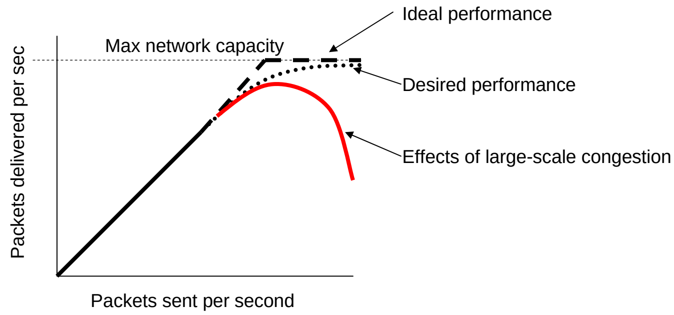
Congestion control

How does congestion arise?



- Router receives data faster than it can send it
  - Downstream link or router congested or failed
  - Extreme traffic on one output link
- Usually a transient condition, but crisis can occur if not relieved quickly
  - Congestion cascades back into upstream routers
  - Whole sections of the network can be blocked quickly

Congestion can lead to catastrophe



The University Of Alabama in Huntsville Computer Science

Congestion indicators

cs570

- At routers:
  - Average queue lengths
  - Number of dropped queue items
  - Ping delay to neighbors
- At senders, receivers:
  - Number of timeouts
  - RTT (measured from data send to ACK receive)
  - Ping delay
  - Receive buffer fullness

Some things to keep in mind

cs570

The University Of Alabama in Huntsville Computer Science

- Generating additional traffic to report congestion makes the situation worse – piggybacking on existing traffic is better
- Short timeouts lead to needless re-transmits, aggravating congestion
- Some parts of the network can be fine at the same time that other parts are crashing

cs570

The University Of Alabama in Huntsville Computer Science

Dealing with congestion

Some Methods for controlling congestion

cs570

- Add resources
- Prevent it (Open-loop methods)
- Detect and Relieve it (Closed-loop methods)

The University Of Alabama in Huntsville Computer Science

Adding resources

cs570

- Spread load over multiple routes
- Activate backup routers, links
- Increase power (satcom)
- Adding resources won't be feasible in many situations – we need to be able to deal with congestion using the resources we have

The University Of Alabama in Huntsville Computer Science

The University Of Alabama in Huntsville Computer Science

cs570

Open-loop methods

G. W. Cox -- Fall 2007

Congestion

The University Of Alabama in Huntsville Computer Science

The idea

cs570

- Prevent congestion from happening
  - Usually, no feedback from the network
- Some approaches:
  - Adjust retransmission policy
  - Adjust ACK policy
  - Spread traffic
  - Adjust TTL usage
  - Improve queuing methods

G. W. Cox -- Fall 2007

Congestion

Retransmission policy

The University Of Alabama in Huntsville Computer Science — **cs570**

- The more retransmissions, the more likely there will be congestion
  - Increasing timeouts tends to reduce congestion
  - Minimizing amount of data retransmitted reduces congestion
    - Selective repeat instead of go-back-n
    - Receiver able to order out-of-order data instead of asking for repeat

Acknowledgement Policy

The University Of Alabama in Huntsville Computer Science — **cs570**

- ACK each packet immediately → extra ACK traffic
- Save ACKs to piggyback on reverse data → more timeouts, so more retransmission traffic

The University Of Alabama in Huntsville Computer Science

Routing approaches

cs570

- Spreading traffic over multiple paths tends to prevent congestion

The University Of Alabama in Huntsville Computer Science

TTL

cs570

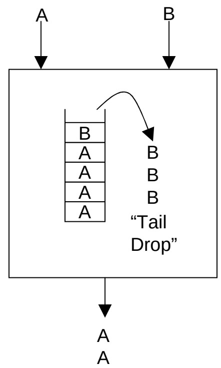
- Long TTL → Lost packets clog up the network
- Short TTL → Less direct clogging, but more timeouts → more retransmissions

Improved queuing

cs570

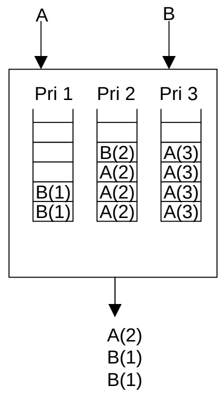
The University Of Alabama in Huntsville   Computer Science

Standard FIFO



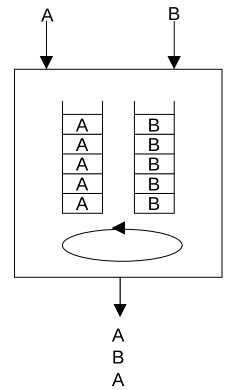
- One user penalized for another's traffic
- No way to recognize hi-priority traffic

Priority FIFO



- Need some way to mark segment or channel priority

"Fair" queue



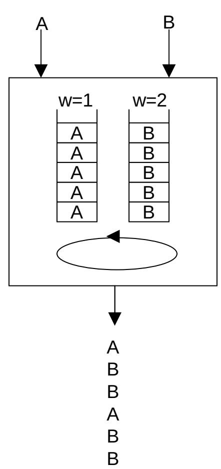
- No way to bias toward a hi-pri channel

Weighted fair queuing

cs570

The University Of Alabama in Huntsville   Computer Science

Combines features of priority and fair queuing



Each queue is assigned a weight that determines the number of queued items that will be taken each output cycle

The University Of Alabama in Huntsville Computer Science

cs570

Closed-Loop Approaches

The University Of Alabama in Huntsville Computer Science

The idea

cs570

- Monitor the network to detect increase in congestion, then act to reduce it
- Some approaches:
  - Allow routers to report congestion to senders
  - Allow routers to drop load
  - Adjust the TCP send window size dynamically

“DEC bit”

cs570

- When a router is nearing overload, it sets a special “congestion bit” in the packets/segments it processes.
- The congestion bit is returned to senders as part of ACKs
- When a sender sees congestion bit set, it can reduce send rate to that destination

Choke packets

cs570

- When a router is nearing overload, it sends a special “choke packet” to the source(s) sending traffic at the moment. The choke packet identifies the destination from the offending packets.
- A source that receives a choke packet reduces the rate of traffic to the destination by a fixed percentage. This lasts for a fixed amount of time.
- An alternate: Choke packets on a hop-by-hop basis

Random Early Deletion (RED)

cs570

- Designed to work with TCP Congestion Window approach
- When router nears overload, it starts randomly deleting "low priority" packets
- This triggers timeouts, which will ultimately cause congestion window size to be reduced.
- How does router determine "low priority"?
  - Randomly (across channels, within a channel)
  - Special flags ("Do Not Delete"…)

The University Of Alabama in Huntsville Computer Science

G. W. Cox -- Fall 2007

Congestion

TCP Congestion Window

cs570

- Original TCP did not include congestion control, ARPAnet nearly collapsed (early '80s) → TCP patched
- The idea: Make TCP slow down send rate as network congestion increases
- Method:
  - Send Window size set at MAX(advertised_size, CW_size)
  - Sending TCP sets CW_size dynamically based on perceived network congestion (as indicated by rate of timeouts)
    - Each time send is successful, CW_size = CW_size + 1
    - Each timeout, CW_size = CW_size/2
  - The effect: slow acceleration of send rate when the network is healthy, slam on the brakes when network starts slowing down ("additive increase, multiplicative decrease")

The University Of Alabama in Huntsville Computer Science

G. W. Cox -- Fall 2007

Congestion

Source-based control (TCP-Vegas)

- Sending TCP determines “no-congestion” RTT (usually determined by ACK for first seg to a destination)
- Dynamic RTT (measured from ACKs) is compared to no-congestion RTT.
- As RTT increases, CW_size is reduced