Congestion Control

Module Goals

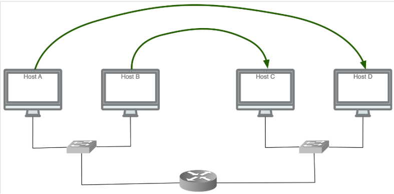
At the conclusion of this module, students will be able to

- define congestion and explain the primary source of congestion
- calculate the fairness of a set of network flows

A Sample Network

- consider two hosts communicating with two other hosts through a router with infinite buffers
  - router can transmit at R bps
  - hosts can transmit at T bps
- three possible cases:

$$T \ll {}^{R}\!/_{2} \quad T \gg {}^{R}\!/_{2} \quad T \approx {}^{R}\!/_{2}$$



Congestion Consequences

- what happens in each situation?
- $T \ll R/2$: if the hosts are sending much less than the output rate of the router, things will be okay
- $T \gg R/2$: if the hosts are sending much more than the output rate, things will be really bad
- $T \approx R/2$: if the hosts are sending just about as fast as the router can output, things will be pretty bad too

How can we model this?

A Quick Aside: M/M/1 Queues

- model of a queue!

  (typically assuming the queue is infinitely sized!)

- time between arrivals follows an exponential distribution:

  $$f_\lambda = \lambda e^{-\lambda t}$$

  with average time $1/\lambda$

- time to service an arrival follows an exponential distribution:

  $$g_\mu = \mu e^{-\mu t}$$

  with average time $1/\mu$

Congestion Consequences

- the average number of items in the queue is

$$L_q = \rho^2/(1 - \rho)$$

  where $\rho = \lambda/\mu$

- if the hosts are sending much less than the output rate of the router, things will be okay

  (if $\lambda \ll \mu$, the queue services arrivals quickly)

- surprisingly, if the hosts are sending just about as fast as the router can output, things are pretty bad too!

  ($\lambda \to \mu$, i.e., $\rho \to 1$, the average number of items explodes towards infinity!)

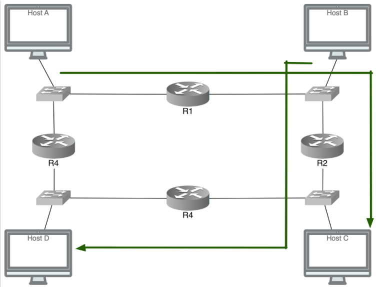
- congestion occurs as packet arrival rate nears link capacity

Costs of Congestion

- consider the same network, but now the router has finite buffers
- as the buffers fill, segments get dropped
- hosts will retransmit dropped segments when discovered
- every time we retransmit something, we are not transmitting something new
- if we only transmit dropped segments, our transmission rate of new segments drops to about $R/3$
- if we erroneously retransmit non-dropped segments, the effective transmission rate drops to about $R/4$

Costs of Congestion

- consider a more complicated network...
- we can starve the A–C connection if the B–D traffic is greater than the output from R1
- if packets from the A–C connection are constantly dropped, the work done by R1 is pointless and upstream capacity is wasted



Our Goal

- if congestion is the problem, congestion control becomes our goal
- to do that, we need to effectively use the network resources available to us:
  - the bandwidth of the links
  - the buffer space at the routers

The Players

- doing this requires effort from both the network and the hosts!
- network responsibilities: various queuing disciplines can be used to control the order in which packets get transmitted and which packets get dropped
- host responsibilities: use congestion control mechanisms to pace how fast sources are allowed to send packets

A Flow-Oriented View

- we're going to assume that the network is essentially connectionless, with any connection-oriented service implemented in the transport protocol
- however, this requires us to update our definition of connectionless a little
- classically, all datagrams are completely independent in a connectionless network

  (datagrams are switched independently, but who sends just one datagram?)
- there is realistically a **flow** of related datagrams through the network

A Flow-Oriented View

- flows can be abstracted at different granularity
  - host-to-host
    (have the same source/destination host addresses)
  - process-to-process
    (have the same source/destination host/port pairs)
- you might hear the later situation called a **channel**
- one last thing: flows can be **explicit** or **implicit**
  - implicit flows are set up ad hoc by routers who just happen to notice a sequence of packets from one host to another
  - explicit flows are set up, but don’t provide the reliability or delivery of a proper connection

Fair Resource Allocation

- a good first step towards controlling congestion would be allocating network resources fairly
- what is “fair”?
  - a reservation-based resource allocation scheme provides an explicit way to create unfairness
  - with such a scheme, we might use reservations to enable a video stream to receive 1 Mbps across some link while a file transfer receives only 10 Kbps over the same link
  - probably not fair, even if it might be effective

Jain's Fairness Index

- Raj Jain[1], proposed a new fairness index to quantify the fairness of a resource allocation scheme:
  - given a set of flows $1, 2, \ldots, n$ with throughputs $x_1, x_2, \ldots, x_n$ (measured in consistent units), Jain's Fairness Index assigns the following index

$$f(x_1, x_2, \ldots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}$$

  - the index always results in a value between 0 and 1, with 1 representing perfect fairness

[1] Professor at Washington University with excellent notes online for networks and network security: `http://www.cse.wustl.edu/~jain/`