Congestion Control and Resource Allocation

Lecture material taken from

"Computer Networks *A Systems Approach*", Third Edition, Peterson and Davie,

Morgan Kaufmann, 2003.

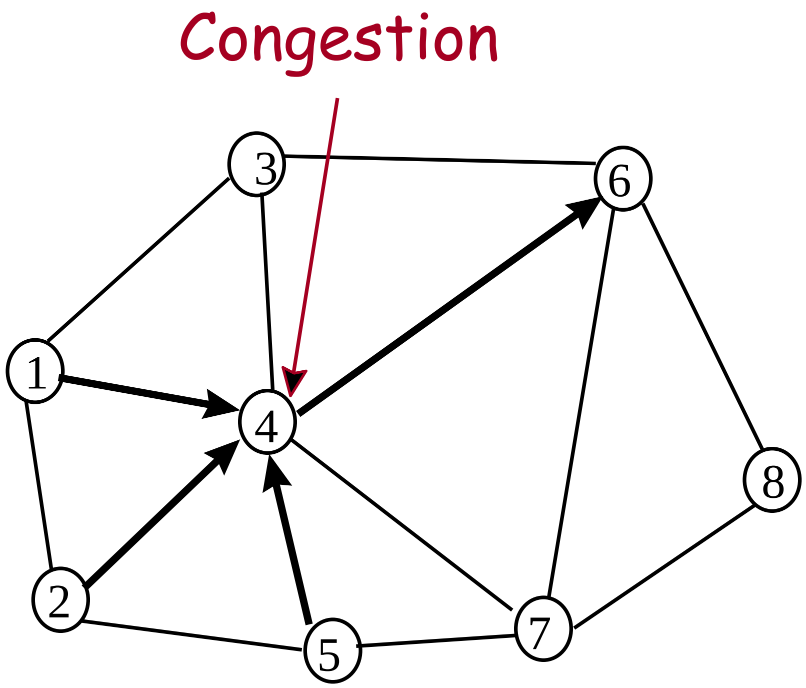
Definitions

- Flow control:: keep a fast sender from overrunning a slow receiver.
- Congestion control:: the efforts made by network nodes to prevent or respond to overload conditions.

Congestion control ***is intended to keep a fast sender from sending data into the network due to a lack of resources in the network {e.g., available link capacity, router buffers}.***

Congestion Control

- Congestion control is concerned with the **bottleneck routers** in a packet switched network.
- Congestion control can be distinguished from routing in that sometimes there is no way to *'route around'* a congested router.



Leon-Garcia & Widjaja: ***Communication Networks***

Figure 7.50b

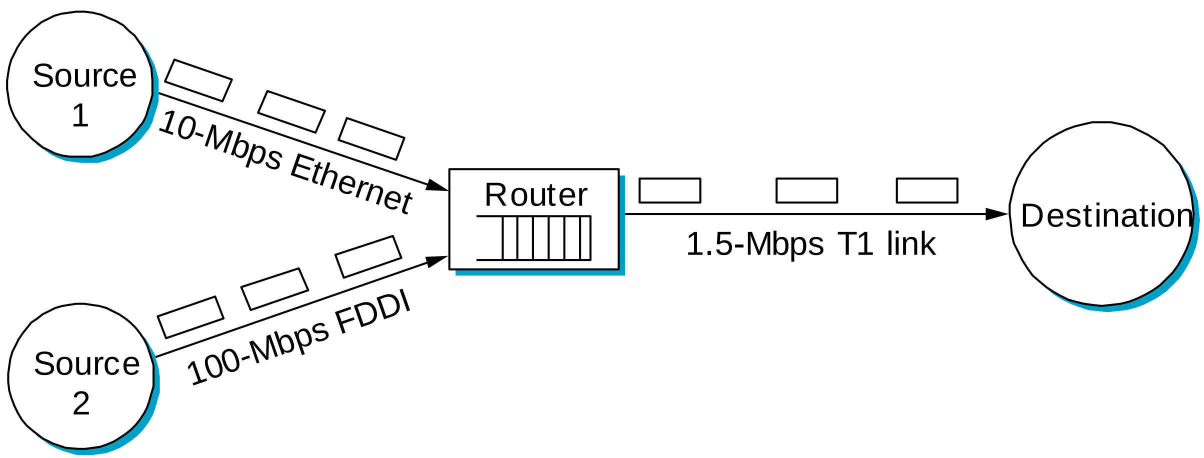


Figure 6.1 Congestion in a packet-switched network

P&D slide

Flows

- **flow ::** a sequence of packets sent between a source/destination pair and following the same route through the network.
- Connectionless flows within the TCP/IP model:: The connection-oriented abstraction, TCP, is implemented at the transport layer while IP provides a connectionless datagram delivery service.
- With connectionless flows, there exists no state at the routers.

Flows

- Connection-oriented flows (e.g., X.25) – connection-oriented networks maintain hard state at the routers.
- ***Soft state* ::** represents a middle ground where *soft state* is not always explicitly created and removed by signaling.
- Correct operation of the network does not depend on the presence of soft state, but soft state can permit the router to better handle packets.

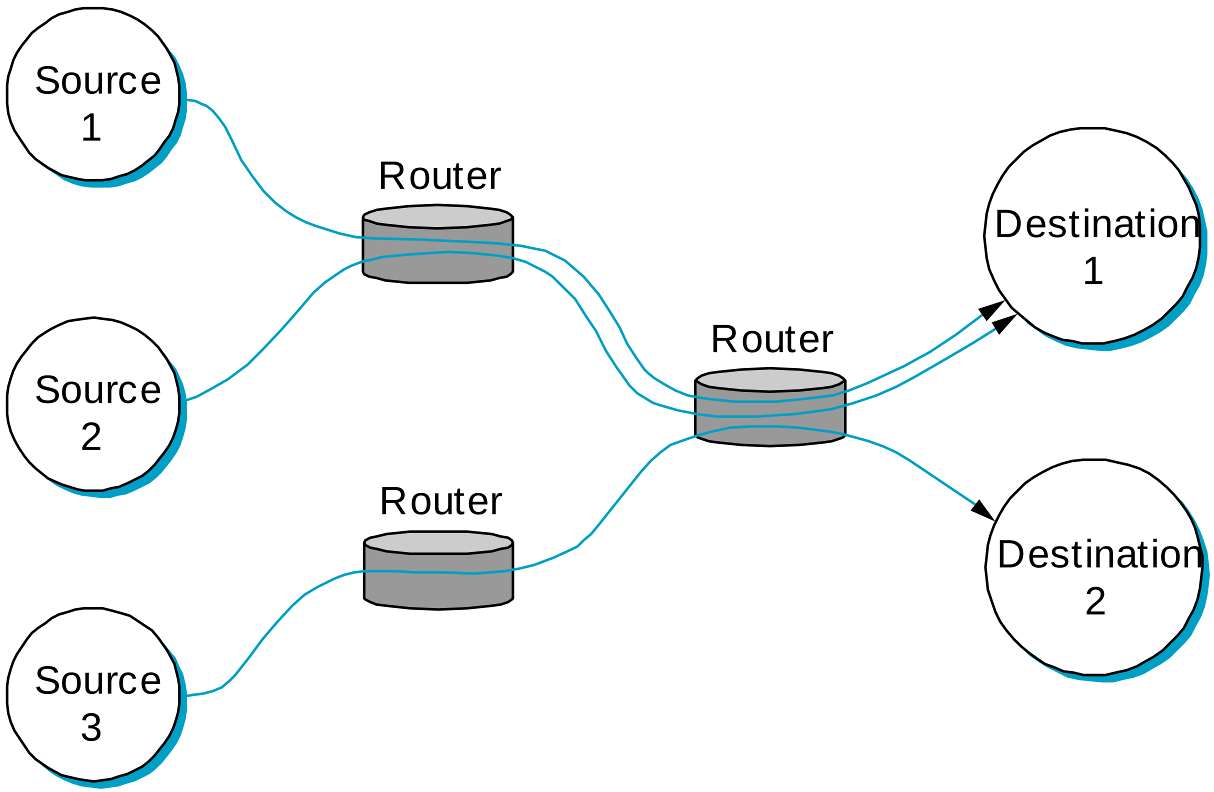


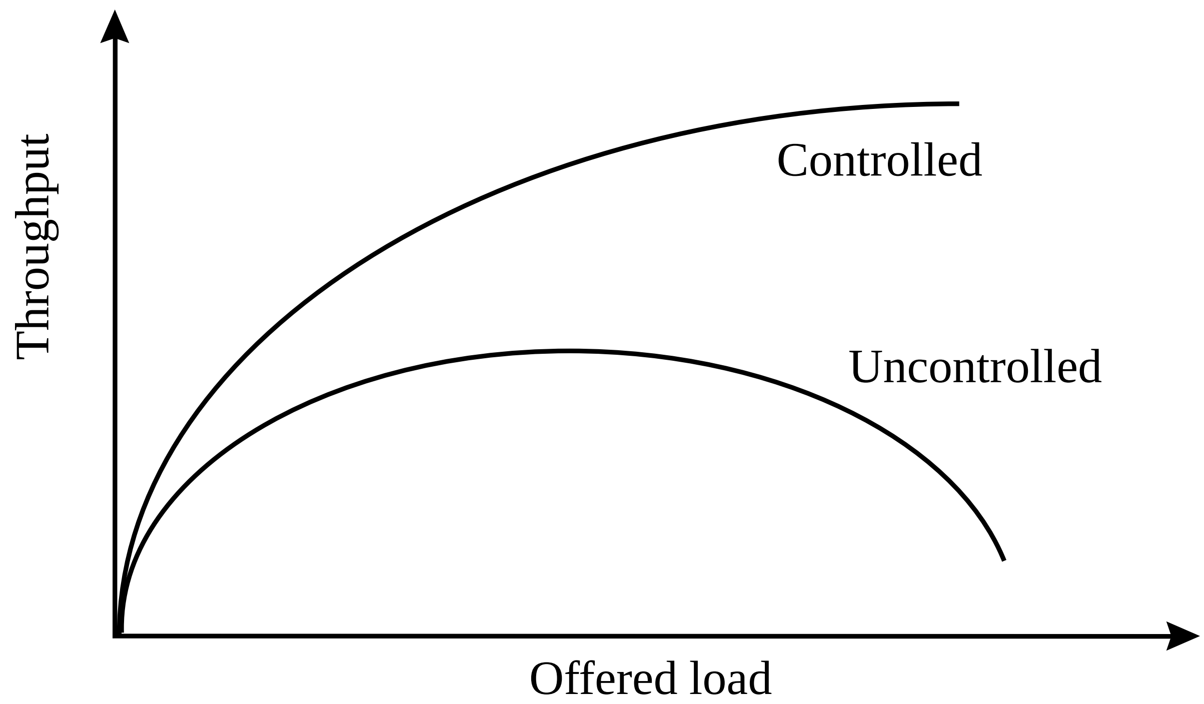
Figure 6.2 Multiple Flows passing through a set of routers

P&D slide

Service

- Best-effort service :: The hosts are given no opportunity to ask for guarantees on a flow's service.
- QoS (Quality of Service) :: is a service model that supports some type of guarantee for a flow's service.

Lack of Congestion Control



Leon-Garcia & Widjaja: *Communication Networks*

Figure 7.51

Congestion Control Taxonomy

- Router-Centric
  - The internal network routers take responsibility for:
    - Which packets to forward
    - Which packets to drop or mark
    - The nature of congestion notification to the hosts.
  - This includes the Queuing Algorithm to manage the buffers at the router.
- Host-Centric
  - The end hosts adjust their behavior based on observations of network conditions.
  - (e.g., TCP Congestion Control Mechanisms)

Congestion Control Taxonomy

- Reservation-Based – the hosts attempt to reserve network capacity when the flow is established.
  - The routers allocate resources to satisfy reservations or the flow is rejected.
  - The reservation can be receiver-based (e.g., RSVP) or sender-based.

Congestion Control Taxonomy

- Feedback-Based - The transmission rate is adjusted (via window size) according to feedback received from the sub network.
  - Explicit feedback – FECN, BECN, ECN
  - Implicit feedback – router packet drops.
- Window-Based - The receiver sends an advertised window to the sender or a window advertisement can be used to reserve buffer space in routers.
- Rate-Based – The sender's rate is controlled by the receiver indicating the bits per second it can absorb.

Evaluation Criteria

- Evaluation criteria are needed to decide how well a network ***effectively*** and ***fairly*** allocates resources.
- Effective measures – throughput, utilization, efficiency, delay, queue length, goodput and power.

$$\text{Power} = \frac{\text{throughput}^{\alpha}}{\text{delay}}$$

Fairness

- Jain's fairness index

For any given set of user throughputs ($x1$, $x2$,…xn), the fairness index to the set is defined:

$$f(x1, x2, \ldots, xn) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n} x_i^2}$$

- Max-min fairness

Essentially 'borrow' from the rich-in-performance to help the poor-in-performance

For example, CSFQ

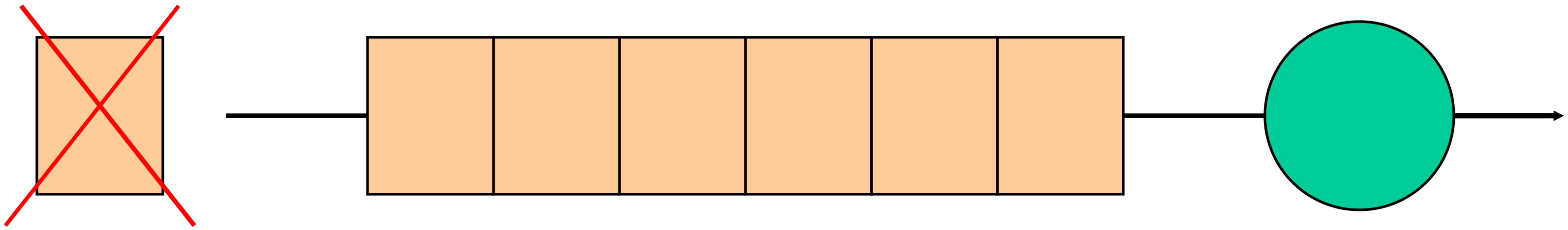
Congestion Control (at the router)

- Queuing algorithms determine:
  - How packets are buffered.
  - Which packets get transmitted.
  - Which packets get marked or dropped.
  - *Indirectly determine the* **delay** *at the router.*
- Queues at outgoing links drop/mark packets to implicitly signal congestion to TCP sources.
- Remember to *separate* queuing policy from queuing mechanism.

Congestion Control (at the router)

- Some of the possible choices in queuing algorithms:
  - FIFO (FCFS) *also called* Drop-Tail
  - Fair Queuing (FQ)
  - Weighted Fair Queuing (WFQ)
  - Random Early Detection (RED)
  - Explicit Congestion Notification (ECN).

Drop Tail Router [FIFO]



- First packet to arrive is first to be transmitted.
- **FIFO** queuing mechanism that drops packets from the *tail of the queue* when the queue overflows.
- Introduces ***global synchronization*** when packets are dropped from several connections.
- **FIFO** is the scheduling mechanism, **Drop Tail** is the policy

Priority Queuing

- Mark each packet with a priority (e.g., in TOS (Type of Service field in IP)
- Implement multiple **FIFO** queues, one for each priority class.
- Always transmit out of the highest priority non-empty queue.
- Still no guarantees for a given priority class.

Priority Queuing

- Problem:: high priority packets can 'starve' lower priority class packets.
- Priority queuing is a simple case of "differentiated services" [DiffServ].
- One practical use in the Internet is to protect routing update packets by giving them a higher priority and a special queue at the router.

Fair Queuing [FQ]

- The basic problem with **FIFO** is that it does not separate packets by flow.
- Another problem with **FIFO** :: an "ill-behaved" flow can capture an arbitrarily large share of the network's capacity.

Idea:: maintain a separate queue for each flow, and Fair Queuing (**FQ**) services these queues in a round-robin fashion.

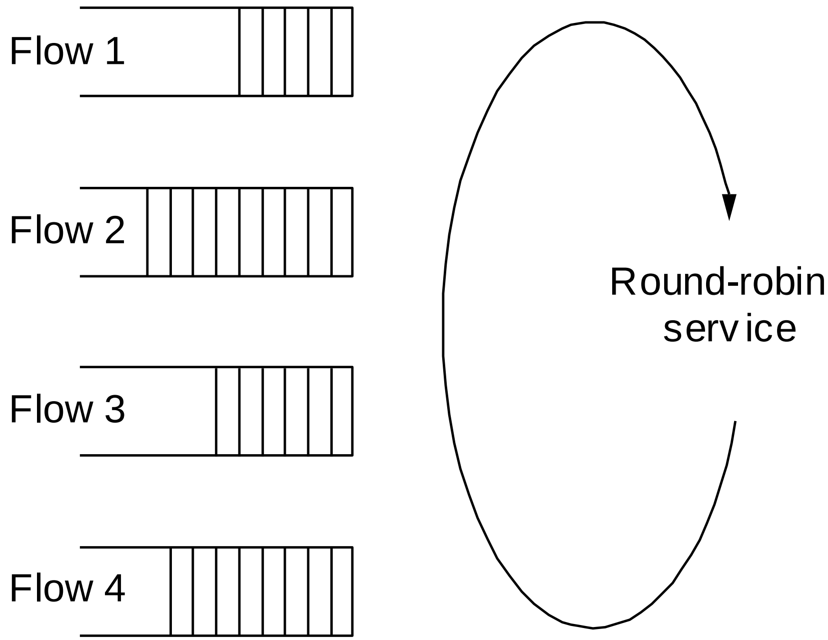


Figure 6.6 Fair Queuing

P&D slide

Fair Queuing [FQ]

- "Ill-behaved" flows are *segregated* into their own queue.
- There are many implementation details for **FQ**, but the main problem is that *packets are of different lengths* ➔ *simple* **FQ** *is not fair!!*
- **Ideal FQ::** ***do bit-by-bit round-robin.***

Fair Queuing [FQ]

- **FQ** simulates bit-by-bit behavior by using timestamps (too many details for here!).
- One can think of **FQ** as providing a guaranteed minimum share of bandwidth to each flow.
- **FQ** is ***work-conserving*** in that the server is never idle as long as there is a customer in the queue.

* **Note:** The **per-flow state information kept at the router is expensive (it does not scale).**

Weighted Fair Queuing [WFQ]

WFQ idea:: Assign a weight to each flow (queue) such that the weight logically specifies the number of bits to transmit each time the router services that queue.

- This controls the percentage of the link capacity that the flow will receive.
- The queues can represent "classes" of service and this becomes DiffServ.
- An issue – how does the router learn of the weight assignments?
  - Manual configuration
  - Signaling from sources or receivers.