ECE/COMPSCI 356 Computer Network Architecture

Lecture 21: Resource Allocation and Congestion Avoidance

Neil Gong

neil.gong@duke.edu

Slides credit: Xiaowei Yang, PD

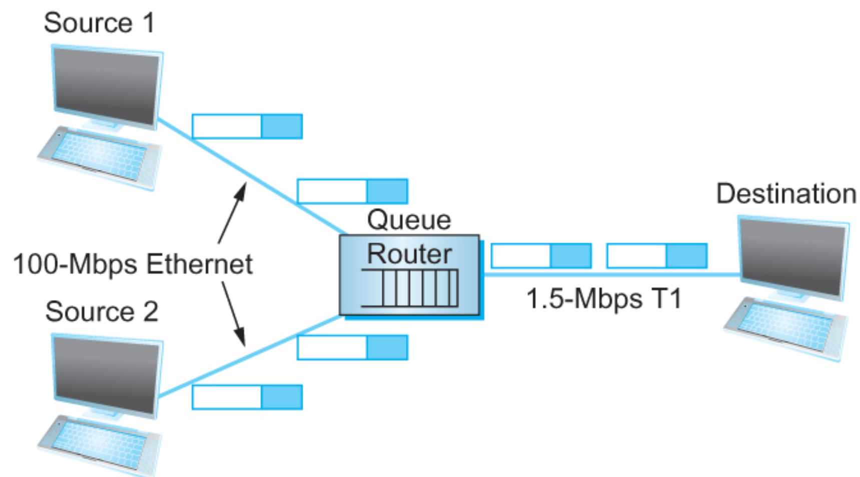
Overview

- Resource allocation
  - Taxonomy
  - Evaluation criteria
  - Queuing disciplines

- Congestion avoidance
  - DEC bit
  - RED
  - Source-based congestion avoidance

Resource Allocation

- Resources
  - Bandwidth of the links
  - Buffers at the routers and switches



- Different TCP connections share resources

- How to allocate resources to different TCP connections?

Taxonomy of Resource Allocation

- Router-centric versus Host-centric
- Reservation-based versus Feedback-based

Router-centric versus Host-centric

- Router-centric
  - router decides when packets are forwarded and dropped
  - router informs hosts how many packets they are allowed to send
- Host-centric
  - hosts observe the network and adjust their behavior
- These two groups are not mutually exclusive

Reservation-based versus Feedback-based

- Reservation-based
  - Host asks the network to reserve resources for a flow
  - Each router allocates resources (buffers and/or percentage of the link’s bandwidth) to satisfy the reservation
  - If the reservation cannot be satisfied at some router, then the router rejects the reservation
- Feedback-based
  - Hosts send data without reserving resources
  - Adjust sending rate according to feedback.
    - explicit feedback: a congested router sends a “please slow down” message to the host
    - implicit feedback: host adjusts sending rate according to timeout.

Evaluation Criteria

- Effective Resource Allocation
  - Maximize network utilization with small delays
- Fair Resource Allocation
  - Fairness among multiple TCP connections

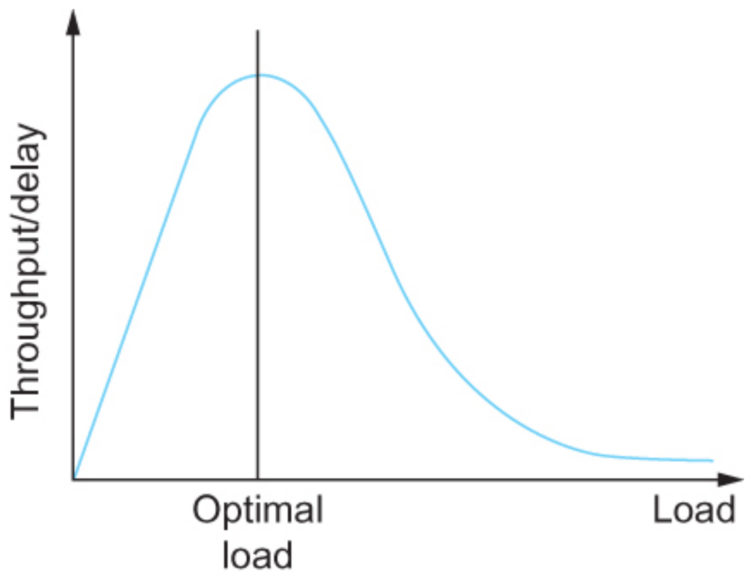
Effective Resource Allocation

- Two principal metrics of networking
  - throughput
  - delay
- Maximize throughput while minimizing delay.
- Trade-off between throughput and delay

Effective Resource Allocation

- Measure the trade-off
  - Ratio of throughput to delay
- Called *power of the network*
- Power = Throughput/Delay

Effective Resource Allocation



Ratio of throughput to delay as a function of load

Fair Resource Allocation

- Consider fairness among multiple TCP connections.
- What is a fairness metric?
  - Raj Jain proposed a fairness metric.
  - Given a set of flow throughputs (x_1, x_2, . . . , x_n) (measured in consistent units such as bits/second), we have the following fairness index:

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

  - The fairness index is between 0 and 1, with 1 representing greatest fairness.

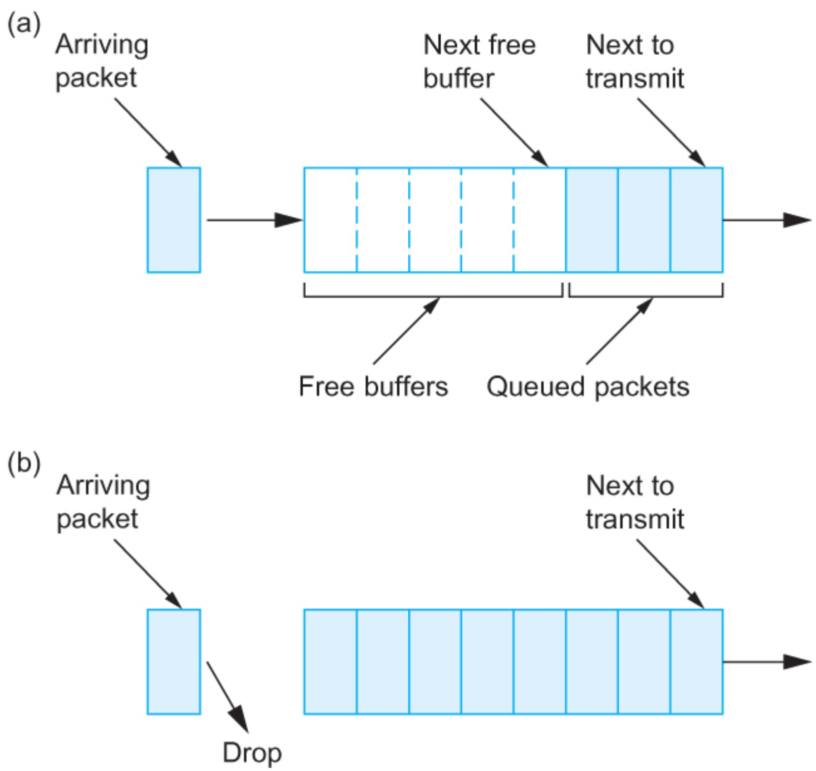
Queuing Disciplines

- How a router manages packets
  - Which one to send next
  - Which one to drop when buffer is full
- Different queuing disciplines
  - FIFO, also called FCFS
  - Priority queuing
  - Fair queuing

FIFO

- Which one to send next
  - The first packet that arrives at a router is the first packet to be transmitted
- Which one to drop
  - If a packet arrives and the queue is full, then the router discards the packet
  - Which flow the packet belongs to is not considered
  - Also called *tail drop,* since packets that arrive at the tail end of the FIFO are dropped

FIFO



(a) FIFO queuing; (b) tail drop at a FIFO queue.

Problems of FIFO

- Priority is not considered
  - Priority Queuing

- Flow is not considered
  - Fair Queuing

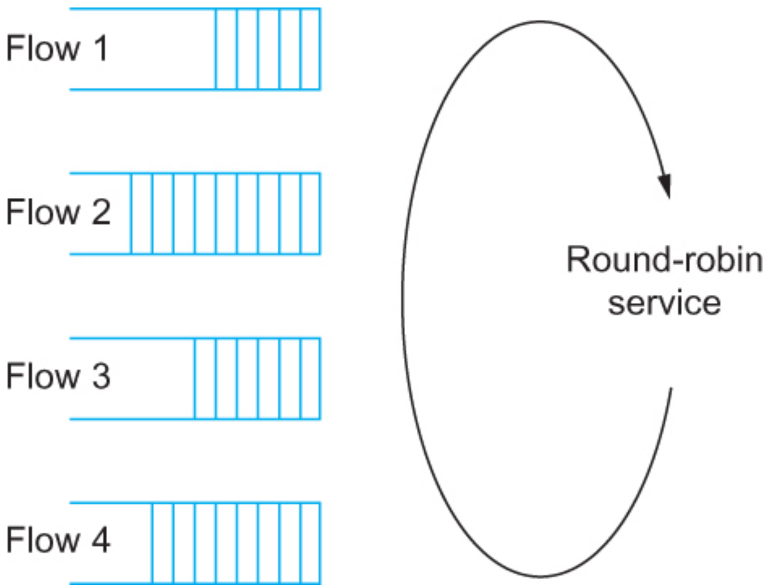
Priority Queuing

- A simple variation of FIFO
- Mark each packet with a priority
  - The mark could be carried, for example, in the IP header.
- The routers implement multiple FIFO queues
  - One for each priority class
  - The router transmits packets out of the highest-priority, non-empty queue
  - Within each priority queue, packets are managed in a FIFO manner.

Fair Queuing

- Basic idea
  - Router maintains a separate queue for each flow
  - Router services these queues in a sort of round-robin

Fair Queuing



Round-robin service of four flows at a router

Fair Queuing

- Challenge
  - Packets are not necessarily the same length.
  - Suppose a router is managing two flows
    - One with 1000-byte packets and the other with 500-byte packets
    - A simple round-robin service gives the first flow two thirds of the link's bandwidth and the second flow only one-third of its bandwidth.

- Solution: Bit-by-bit round-robin
  - Router transmits a bit from flow 1, then a bit from flow 2, and so on.

Fair Queuing

- However, it is not feasible to interleave the bits from different packets.
- Solution
  - Fair Queuing simulates this behavior
  - First determine when a given packet would finish being transmitted if it were being sent using bit-by-bit round-robin
  - Use this finishing time to sequence the packets for transmission

Fair Queuing

- Image a clock
  - ticks when a bit from all active flows is transmitted
- Consider the behavior of a single flow
- For this flow, let
  - P_i : denote the length of packet i
  - S_i: time when the router starts to transmit packet i
  - F_i: time when router finishes transmitting packet i
  - Clearly, $F_i = S_i + P_i$

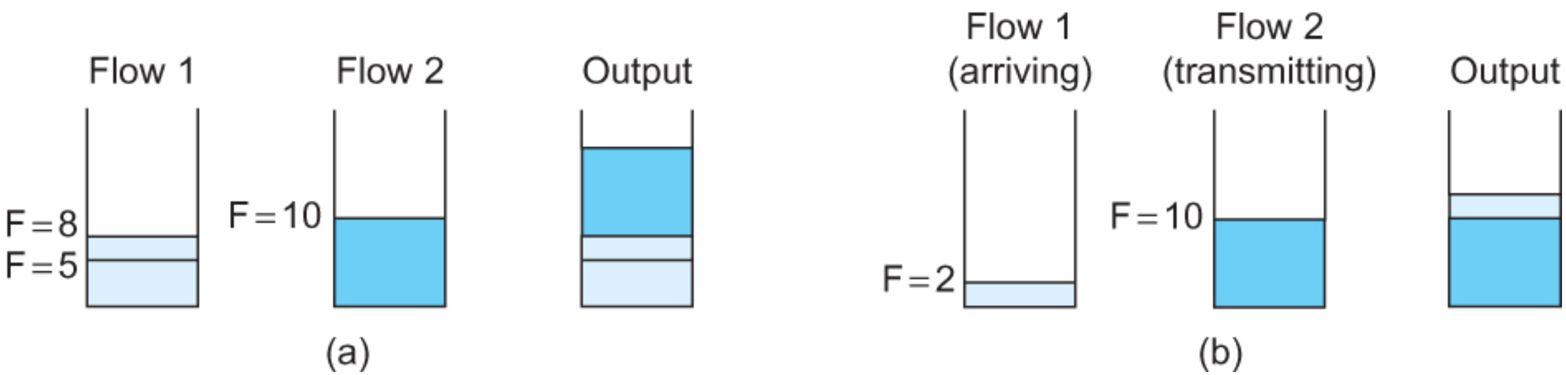
Fair Queuing

- When do we start transmitting packet i?
  - Depends on whether packet i arrived before or after the router finishes transmitting packet i-1 for the flow
- Let A_i denote the time that packet i arrives at the router
- Then $S_i = \max(F_{i-1}, A_i)$
- $F_i = \max(F_{i-1}, A_i) + P_i$

Fair Queuing

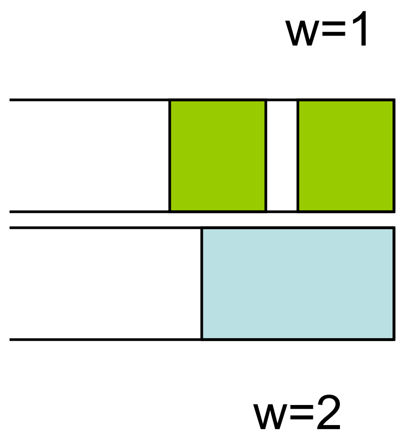
- For every flow, we calculate F_i for each packet that arrives using our formula
- We treat all the F_i as timestamps
- Next packet to transmit is always the packet that has the lowest timestamp
  - The packet that should finish transmission before all others

Fair Queuing – an example



Example of fair queuing in action: (a) packets with earlier finishing times are sent first; (b) sending of a packet already in progress is completed

Weighted Fair Queuing



- Different queues get different weights
  - Take w_i amount of bits from a queue in each round
  - $F_i = S_i + P_i / w_i$

Overview

- Resource allocation
  - Taxonomy
  - Evaluation criteria
  - Queueing disciplines
- **Congestion avoidance**
  - DEC bit
  - RED
  - Source-based congestion avoidance

Congestion Control vs. Avoidance

- Congestion control
  - Host repeatedly increases sending speed until congestion happens
  - Then decrease sending speed
  - Adopted by TCP
- Congestion avoidance
  - Predict when congestion is about to happen and then reduce sending speed before congestion happens.
  - Not widely adopted yet

DEC Bit

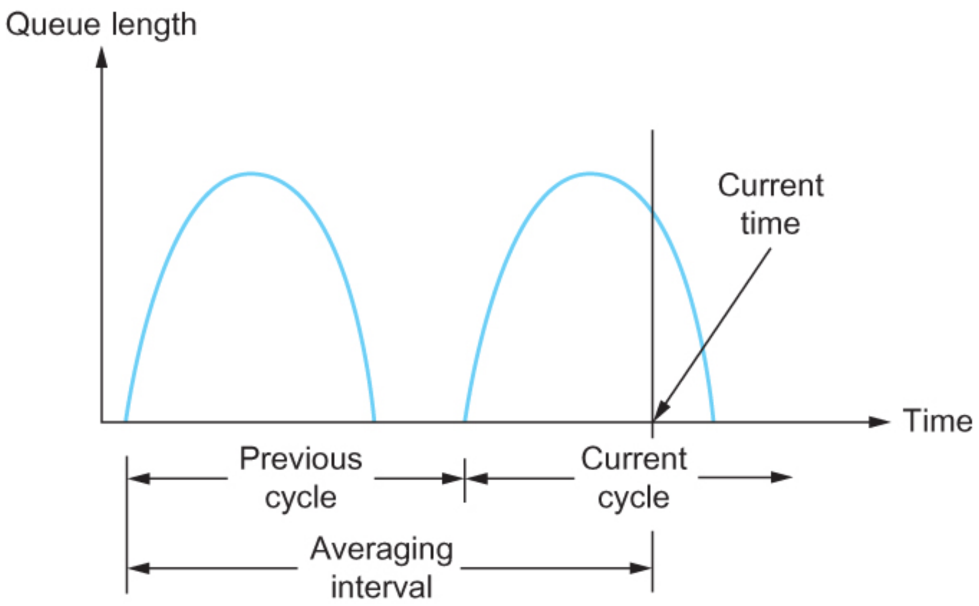
- Each router monitors its load and notifies the end nodes when congestion is about to occur.
- Notification is implemented by setting a binary congestion bit in the packets that flow through the router; hence the name DEC bit.
- Destination host copies this congestion bit into the ACK it sends back to the source.
- Source adjusts its sending rate to avoid congestion

DEC Bit – when a router sets it?

- A router sets this bit in a packet if its average queue length is greater than or equal to 1 at the time the packet arrives.

- Average queue length is measured over a time interval that spans the last busy+idle cycle, plus the current busy cycle.

DEC Bit



Computing average queue length at a router

DEC Bit – when a source adjusts speed?

- Source maintains a congestion window, like TCP, and calculates what fraction of the last window's worth of packets resulted in the bit being set.
- Variant of AIMD
  - If less than 50% of the packets had the bit set, then the source increases its congestion window by one packet.
  - If 50% or more of the packets had the congestion bit set, then the source decreases its congestion window to 0.875 times the previous value.

Random Early Detection (RED)

- Instead of explicitly sending a congestion notification to source, RED *implicitly notifies* source by dropping one of its packets.
  - The source is effectively notified by the subsequent timeout or duplicate ACK.

- When to drop a packet and what packet to drop?

Random Early Detection (RED)

- Consider a simple FIFO queue.
- *RED* drops each arriving packet with some *drop probability* when the queue length exceeds some *drop level.*
- How to decide drop level and drop probability

Random Early Detection (RED)

- First, RED computes an average queue length using a weighted running average.
  - $\text{AvgLen} = (1 - \text{Weight}) \times \text{AvgLen} + \text{Weight} \times \text{SampleLen}$
  - where $0 < \text{Weight} < 1$ and SampleLen is the length of the queue when a sample measurement is made.
- Software implementation
  - SampleLen is measured every time a new packet arrives.
- Hardware implementation
  - SampleLen is calculated at fixed sampling interval.

Random Early Detection (RED)

- Second, RED has two queue length thresholds
  - MinThreshold
  - MaxThreshold
- When a packet arrives, RED compares the current AvgLen with these two thresholds, according to the following rules:
  - if AvgLen ≤ MinThreshold
    - → queue the packet
  - if MinThreshold < AvgLen < MaxThreshold
    - → calculate probability P (called drop probability)
    - → drop the arriving packet with probability P
  - if MaxThreshold ≤ AvgLen
    - → drop the arriving packet

Random Early Detection (RED)

- P is a function of both AvgLen and how long it has been since the last packet was dropped.

- P is computed as follows:
  - TempP = MaxP × (AvgLen − MinThreshold)/(MaxThreshold − MinThreshold)
  - P = TempP/(1 − count × TempP)
  - Count is the number of non-dropped packets since last drop
    - Count is smaller than a max_count so 1 − count × TempP > 0 & P <=1
    - Or set P as 1 if 1 − count × TempP <=0 or TempP > 1 − count × TempP

Source-based Congestion Avoidance

- General idea
  - Source observes the network behavior and adjusts sending rate when congestion is about to occur
  - E.g., a measurable increase in RTT.
- One Algorithm
  - Every two round-trip delays the algorithm checks to see if the current RTT is greater than the average of the minimum and maximum RTTs seen so far.
  - If yes, decreases the congestion window by one-eighth.

Summary

- Resource allocation
  - Taxonomy
  - Evaluation criteria
  - Queuing disciplines
- Congestion avoidance
  - DEC bit
  - RED
  - Source-based congestion avoidance