Lecture 18: Buffering & Scheduling

CSE 123: Computer Networks
Alex C. Snoeren

HW 3 due next FRIDAY
Lecture 18 Overview

- Queues in routers/switches
- Buffer Management
  - FIFO
  - RED
- Traffic Policing/Scheduling
Third Generation Routers

- Shared interconnect (frequently crossbar)
- Centralized scheduler
- Full forwarding table in line card
- Fixed cells

CSE 123 – Lecture 18: Buffering & Scheduling
Output queuing

- Output interfaces buffer packets

- Pro
  - Simple algorithms
  - Single congestion point

- Con
  - N inputs may send to the same output
  - Requires speedup of N
    » Output ports must be N times faster than input ports
Input queuing

- Input interfaces buffer packets

- Pro
  - Single congestion point
  - Simple to design algorithms

- Con
  - Must implement flow control
  - Low utilization due to Head-of-Line (HoL) Blocking
Head-of-Line Blocking
IQ + Virtual Output Queuing

- Input interfaces buffer packets in per-output virtual queues

- Pro
  - Solves blocking problem

- Con
  - More resources per port
  - Complex arbiter at switch
  - Still limited by input/output contention (scheduler)
Switch scheduling

- **Problem**
  - Match inputs and outputs
  - Resolve contentions, no packet drops
  - Maximize throughput
  - Do it in constant time…

- If traffic is uniformly distributed it's easy
  - Lots of algorithms (approximate matching)

- Seminal result (Dai et al, 2000)
  - Maximal size matching + speedup of two guarantees
  - 100% utilization for most traffic assumptions
Typical high-performance router

- IQ + VoQ + OQ
  - Speedup of 2
  - Central scheduler
  - Fixed-sized internal cells
- Pro
  - Can achieve utilization of 1
  - Can scale to > Tb/s
- Con
  - Multiple congestion points
  - Complexity
Key Router Challenges

- **Buffer management**: which packet to drop when?
  - We only have finite-length queues
- **Scheduling**: which packet to transmit next?
Basic Buffer Management

- FIFO + drop-tail
  - Simplest choice
  - Used widely in the Internet
- FIFO (first-in-first-out)
  - Implies single class of traffic
- Drop-tail
  - Arriving packets get dropped when queue is full regardless of flow or importance
- Important distinction:
  - FIFO: scheduling discipline
  - Drop-tail: drop policy
FIFO/Drop-Tail Problems

- Leaves responsibility of congestion control completely to the edges (e.g., TCP)
- Does not separate between different flows
- No policing: send more packets → get more service
- Synchronization: end hosts react to same events
Active Queue Management

- Design active router queue management to aid congestion control

- Why?
  - Router has unified view of queuing behavior
  - Routers see actual queue occupancy (distinguish queue delay and propagation delay)
  - Routers can decide on transient congestion, based on workload
Design Objectives

- Keep throughput high and delay low
  - High power (throughput/delay)

- Accommodate bursts

- Queue size should reflect ability to accept bursts rather than steady-state queuing

- Improve TCP performance with minimal hardware changes in router
Random Early Detection

- Detect incipient congestion
- Assume hosts respond to lost packets
- Avoid window synchronization
  - Randomly mark packets
- Avoid bias against bursty traffic
Maintain running average of queue length in router

- If $\text{avg} < \text{min}_{th}$ do nothing
  - Low queuing, send packets through

- If $\text{avg} > \text{max}_{th}$, drop packet
  - Protection from misbehaving sources

- Else drop/make packet in a manner proportional to queue length
  - Notify sources of incipient congestion
  - Dropping vs Marking tradeoff (Explicit Congestion Notification)
RED Operation

Max thresh

Min thresh

Average Queue Length

P(drop)

1.0

max_p

min_th

max_th

Avg queue length

P(drop)
For next time...

- NO CLASS MONDAY
- Read Ch. 6.3-4 in P&D
- Homework due next Friday