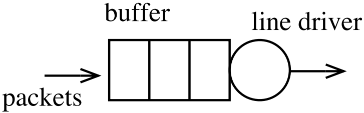
Supporting Loss Guarantees in Buffer-Limited Networks

Mahmoud Elhaddad, Rami Melhem, Taieb Znati

Department of Computer Science
University of Pittsburgh

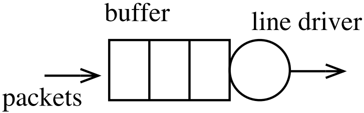
IWQoS 2006

Recent Attention to Router Buffer-Sizing Problems



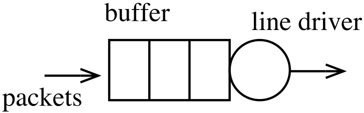
- How small can the buffer be at a link without violating some service requirement?

Recent Attention to Router Buffer-Sizing Problems



- How small can the buffer be at a link without violating some service requirement?
- **Motivation**: on-chip and optical packet buffers
- **Objectives**: bottleneck utilization given TCP, FCFS/DT
- **Results**: Good utilization of a single bottleneck when each flow contribute a small fraction of the load **[sigcomm 04, infocom 06]**

Recent Attention to Router Buffer-Sizing Problems



- How small can the buffer be at a link without violating some service requirement?
- **Motivation**: on-chip and optical packet buffers
- **Objectives**: bottleneck utilization given TCP, FCFS/DT
- **Results**: Good utilization of a single bottleneck when each flow contribute a small fraction of the load **[sigcomm 04, infocom 06]**
- May not be the right problem to solve!
  - restrictive assumptions about traffic may change
  - half the story: loss rate matters

Fixed-Buffer Problems

- Common objectives:
  - maximize network throughput (utilization)
  - provide loss guarantees
- Loss at a link depends on load, arrival process, variability in packet sizes, and the scheduling algorithm

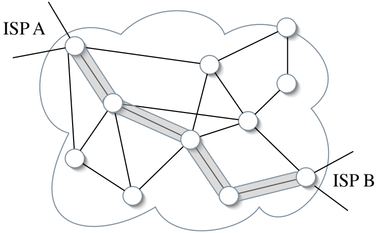
Fixed-Buffer Problems

- Common objectives:
  - maximize network throughput (utilization)
  - provide loss guarantees
- Loss at a link depends on load, arrival process, variability in packet sizes, and the scheduling algorithm
  - shaping to reduce burstiness and loss
  - packing packets into constant size data frames

Fixed-Buffer Problems

- Common objectives:
  - maximize network throughput (utilization)
  - provide loss guarantees
- Loss at a link depends on load, arrival process, variability in packet sizes, and the scheduling algorithm
  - shaping to reduce burstiness and loss
  - packing packets into constant size data frames
  - scheduling: this paper
    - a link scheduling algorithm (service discipline and drop policy)
    - Objective: provide loss guarantees over a multihop network without severly restricting network utilization

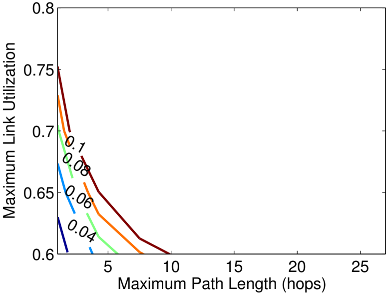
Elements of Network Environment



- Routers connected via time slotted links
- Connections represent ingress–egress traffic aggregates
  - Traffic between two subscribers' networks
  - Fixed bandwidth demands
  - Require loss guarantees: bound on expectation of loss rate

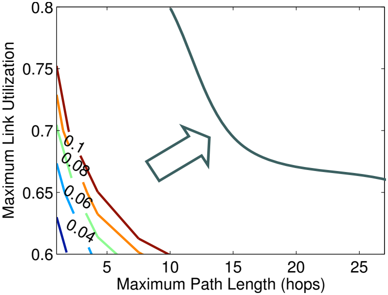
Routing Tradeoffs

- Every Scheduling algorithm defines constraints on max routing-path length and max link utilization
  - so that guarantees are met for all connections



Routing Tradeoffs

- Every Scheduling algorithm defines constraints on max routing-path length and max link utilization
  - so that guarantees are met for all connections

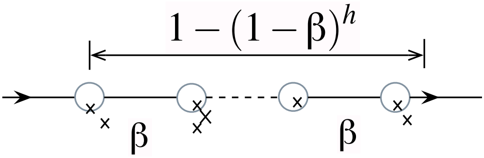


Reservation-based Architecture

- Ingress regularly reserves future slots along path
- Acked reservations act as transmission tokens
- Reservation packets sent over dedicated channels (à la OBS)
- Advantages:
  - virtual lossless pipe of capacity equal to:
    nominal BW − blocking rate (with adequate buffering)
  - Fair sharing of pipe capacity among microflows possible at ingress
- Note: loss rate is equal to reservation request blocking rate when connection fully utilizing its BW allocation

Motivation:

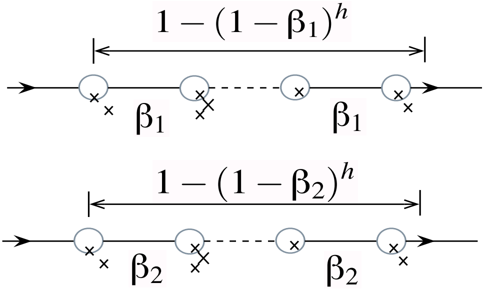
- Consider a connection traversing a sequence of h links
  - Each link treats all requests within a connection identically
  - At every link loss rate = loss probability = β (bernoulli trial)



- Loss rate increases quickly with path length
- Can be viewed as model for FCFS/DT

Idea: Rolling Priority

- Observations
  - Loss rate at a link is the same for all work-conserving algo.
  - No gain in distributing loss in an unfair manner if guarantees apply to all connections
- Example: Look at every pair of consecutive packets (pairs disjoint)

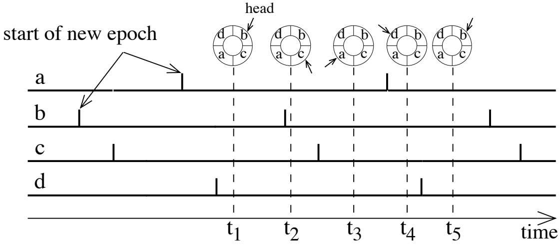


Idea: Rolling Priority

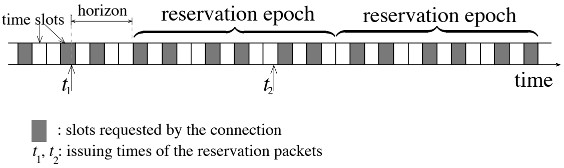
- case 1: Identical treatment $\beta_1 = \beta_2 = 0.5$
  - Expected number of hops until loss of **each** packet: 2.
- case 2: $\beta_1 = 0.9, \beta_2 = 0.1$
  - 50% of packets expected to be lost at 10th hop
  - loss rate for each connection is still 0.5
  - key: consistent assignment of priority to packets throughout path

Idea: Rolling Priority

- From perspective of each connection, time divided into epochs of fixed duration
- new epoch starts periodically
- At each time slot, every link gives higher priority to the connection with earlier starting current epoch

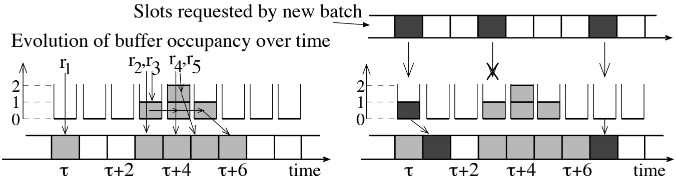


Generation of time slot requests at ingress



- The time between reception of reservation packet at a link and start of corresponding epoch is equal to horizon when reservation packet processing delay is zero.

Contention resolution: Service and drop policies



Analysis Highlights

- the distribution of connection "rank" at a time slot depends on position within epoch
  - Probability concentrated around the mean (consistent priority throughout path–key to good performance)
- The mean rank of a connection drops linearly through epoch
- Connections see load only due to connections of higher rank
- Bounds on blocking probabilities when connections have periodic request streams
- Worst-case routing and bandwidth allocations
- Analyze blocking at each link in isolation
  - Multiplexer of periodic streams
- in terms of buffer occupancy and busy period length in an nD/D/1 queue
  - Blocking probability drops superlinearly across epoch

Sample Results

- Connections are periodic request streams, $B = 5$

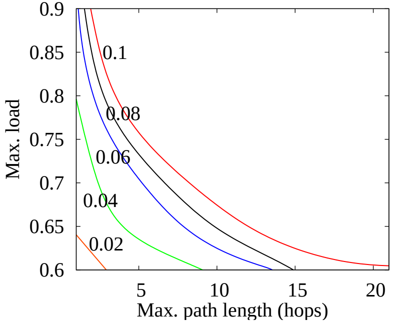


Figure: Left: (a) FCFS/DT packet scheduling, and right: (b) BATCH

Summary

- Fixed-buffer problems are important to look at given the technological constraints on packet buffer implementations
- Reservation-based architecture to provide loss guarantees at flow level and to prevent random loss (important for TCP).
- Introduced a scheduling algorithm called BATCH for improving routing tradeoffs necessary to provide guarantees compared to FCFS/DT
- Analytically obtained loss rate bounds under simplifying assumption
- Validated bounds using simulation

Alternative Scheduling Algorithms

- Globally coordinate transmissions
- Give priority to connection that suffered too much loss upstream
  - Approximations: detecting and preventing streaks
  - Problematic ties: too many connections with loss upstream
- Heuristics: LIS, FTG
  - Unfairness toward a subset of connection. Hard to characterize performance without assumptions on traffic patterns
  - Scheduling algorithm should not give fixed priority to connections at any link