Competitive Analysis of Buffer Policies with SLA Commitments

Boaz Patt-Shamir, Tel Aviv University
Gabriel Scalosub, University of Toronto
Yuval Shavitt, Tel Aviv University
Motivation

• Service Level Agreements (SLA)
  – ATM, DiffServ, MPLS, Metro Ethernet
  – Rate meters
  – Admissible traffic: Token Bucket envelope
  – Additional traffic

• “Show me the money!”
  – SLA violation – costly!
  – Forwarding “out of contract” traffic: More Money!

• Issues:
  – Buffer provisioning, admission control, scheduling
Model

- Single FIFO Queue:
  - Outgoing Rate $r_Q$
  - Buffer size $B_Q$
- Adversarial Traffic:
  - Committed (green):
    - Rate $r \leq r_Q$
    - Burst size $B \leq B_Q$
  - Excess (yellow):
    - Arbitrary
- Also allows best-effort / aggregate

At most $r(I + B)$ green packets in any interval $I$
Model (cont)

• Main constraint (feasibility):
  – All committed traffic must be forwarded

• Discrete time
  – Delivery substep
    • At most $r_Q$ delivered
  – Arrival substep
    • Packets arrive
    • Some yellow packets may be dropped
    • Packets accommodated in the buffer
Metric and Methodology

• Goal:
  Maximize the number of excess packets delivered

• Competitive Analysis:
  Algorithm $A$ is $c$-competitive if for every input sequence $\sigma$
  $$A(\sigma) \geq c \cdot \text{OPT}(\sigma)$$

• Resource augmentation:
  – Buffer size: $\text{OPT}$ uses $B$ whereas $A$ uses $(1 + \epsilon)B$
  – Rate: $\text{OPT}$ uses $r$ whereas $A$ uses $s \geq r$
Our Results

• Lower bounds:
  – Buffer resource augmentation is essential
  – Using $\varepsilon \in [0, 1]$ times more buffer:
    cannot be better than $\varepsilon$-competitive

• Online algorithm ON
  – $\min \left\{ \frac{\varepsilon}{1+\varepsilon - \frac{B - R}{B}}, 1 \right\}$-competitive

• Simulation study:
  – ON is close to optimal
  – Specifically, better than common policies
Previous Work

- Protective buffer management
  - Protective ~ feasibility
  - Push-out
  - Same link speed
  - No analytic guarantees

- Multi-valued packets
  - Const. competitive for finite values

- Packet color marking
  - Exploiting TCP characteristics (AQM)

[Cidon et al. ‘94]

[Kesselman et al. ‘04] [Englert&Westerman ‘06]

[Chait et al. ‘05]
If we use the same amount of buffer as $\text{OPT}$ we can never afford to forward excess.
Upper Bounds

• Lower bounds $\Rightarrow$ buffer resource augmentation
  – Use $(1 + \varepsilon)B$

• Naïve approach:
  – Maintain two queues
  – Give priority to committed queue

• Simulator SIM
  – Same buffer size and rate as $\text{OPT}$
  – Ignores all yellow packets
  – Bounds buffer occupancy of $\text{OPT}$ (by feasibility...)

This is not FIFO
The Concept of Lag

- Lag of a green packet
  \[ \text{lag}^A_t(p) = \max \{ d^A_t(p) - d^\text{SIM}_t(p), 0 \} \]

- \( \varepsilon \)-lag property
  - No green packet in the buffer has lag greater than \( \varepsilon B \)

- Lag of an algorithm
  \[ \phi(t) = \max_{p \in \text{Buff}_A(t)} \text{lag}^A_t(p) \]
Algorithm

Algorithm ON

upon the arrival of a new packet:
1) If yellow: accept if there’s room
2) If green:
   • Drop as few yellow packets from the tail such that the new packet will have lag at most $\varepsilon B$
   • Accept packet

• Algorithm satisfies:
  – Feasibility
  – $\varepsilon$-lag property
Analysis in a Nutshell

• Identify “reset” events:

\[ \phi(t) = 0 \]

• “Overflow” (yellow packets dropped) occurs:
  – Between reset events
  – At least \( \varepsilon B \) yellow packets are “safe” since previous reset
  – Many green packets accepted by SIM:
    • OPT must deal with them as well!!
    • Has “little” space/rate to deal with too many yellow

• Follow algorithm’s lag-difference

\[ \phi(t) - \phi(t - 1) \]
Analysis in a Nutshell (cont)

- Implementation issues:
  - Lag calculation is easy
  - No push-out. Just tail-drop.
Simulation Study

• Bursty SLA-compliant traffic
  – MMPP
  – Color marking (token-bucket)

• Best-effort traffic
  – zero-rate commitment
  – Poisson

• Threshold algorithm
  – Accept yellow packet iff buffer occupancy is below $T$

• OPT upper bound
  – The naïve 2-queue
Simulation Results

- Single MMPP source
- Yellow packets at bursts “tail”
- Yellow traffic: ~ 30% of total traffic

![Graph showing throughput vs. buffer size increase](image)
Simulation Results

- MMPP + Yellow Poisson
- Yellow packets also during OFF
- Yellow traffic: ~ 40% of total traffic
Simulation Results

- MMPP + Yellow Poisson
- Yellow packets also during OFF
- Yellow traffic: ~ 50% of total traffic

![Graph showing throughput vs. buffer size increase for online vs. optimal policies.](image)
Summary

• Algorithm for managing buffers with committed traffic

• Analytic performance results
  – Globally applicable
  – Both lower and upper bounds
  – Guidelines for buffer provisioning

• Simulation study
  – Aggregate flows (w best-effort)
  – Outperforms common approaches
Future Work

• Gaps:
  – No lower bound for large $\varepsilon$.
  – Lower bound vs. upper bound for small $\varepsilon$.

• Multiple queues

Any guesses?
(Recommendation: read the paper first...)
Thank You!