Network QoS
371-2-0213

Lecture 4

Gabriel Scalosub

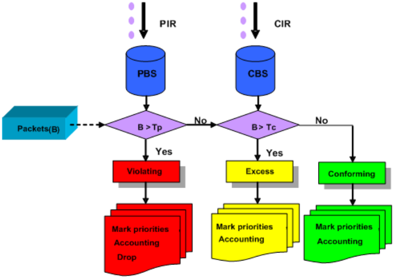
Outline

DiffServ - Key Components

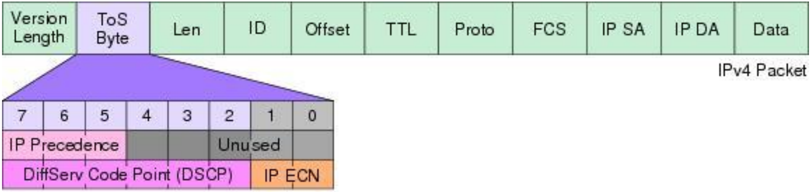
- Main motivation: IntServ handicaps
  - per-flow state
  - complex reservation mechanisms
  - requires large-scale deployment
- Main concern: being "better than best-effort"
- Main design concept: locality of decision
- Key components:
  - middle ground between IntServ and best-effort
  - divide traffic into small number of *forwarding classes*
    - E.g., Gold, Silver, Bronze, Best-Effort
    - class encoded in packet IP header
    - resources allocated per-class (aggregate)
  - based on Service Level Agreements (SLAs)
  - mechanisms
    - marking
    - scheduling (priority queuing, WFQ)
    - buffer management (AQM/RED)

SLAs and Marking

- Specification of expected traffic per-class
  - token-bucket envelope
    - Committed/Peak Information Rate (CIR/PIR)
    - Committed/Peak Burst Size (CBS/PBS)
- Three color marking
  - Done using a dual token bucket
  - E.g., two-rate three color marking



DiffServ Codepoint (DSCP)



- Default PHB:
  - DSCP: $<000000>$ (best-effort)
- Assured Forwarding (AF) PHBs:
  - DSCP: $<001\text{xx}0>$, $<010\text{xx}0>$, $<011\text{xx}0>$, $<100\text{xx}0>$
  - at least two different forwarding classes
  - per class bandwidth allocation
    - implemented using scheduling (e.g., WFQ)
    - each link can determine its own allocation
  - 3 drop priorities per class (e.g., via RED)
- Expedited Forwarding (EF) PHB:
  - DSCP: $<101110>$
  - high-priority queue (10-30% of link capacity)

Competitive Analysis - Definition

- Given an instance I of an optimization problem $\mathcal{P}$, denote by $\mathrm{OPT}(I)$ the value of an optimal feasible solution for I.
- In the online setting, the input to the problem is made available in parts.
- An online algorithm A is said to be c-competitive for problem $\mathcal{P}$ if for every instance I of $\mathcal{P}$, $A(I)$ satisfies:
  - $A(I) \leq c \cdot \mathrm{OPT}(I) + \alpha$ (minimization problem $\mathcal{P}$)
  - $A(I) \geq \frac{1}{c} \cdot \mathrm{OPT}(I) - \alpha$ (maximization problem $\mathcal{P}$)

  where $\alpha \geq 0$ is some additive term independent of I.
- Common to assume that I is generated by an *adversary*
- The online problem is then viewed as a *game*:
  - Adversary (produces I and an optimal solution to I), vs.
  - Algorithm

Model

- Provider's viewpoint: fulfill its end of the SLA
- Focus on a single AF PHB class
- Traffic model
  - committed traffic (green): (r, B) token envelope
  - excess traffic (yellow): arbitrary
  - interleaved
- Queue model
  - single FIFO queue Q
    - buffer of size $B_Q \geq B$
    - service rate $r_Q \geq r$
    - preemptive: may drop enqueued packets

Feasibility

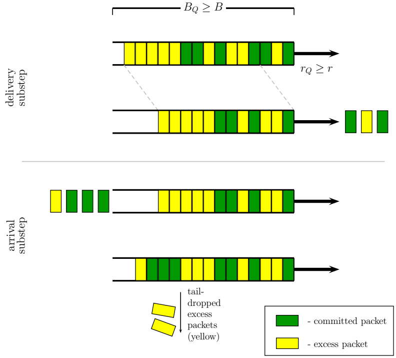
Never drop committed packets

Goal

Maximize number of excess packets forwarded

Time Model and Simplifying Assumptions

- Slotted time
  - delivery/forwarding substep: $\leq r_Q$ forwarded
  - arrival substep: packets may be dropped/accepted



- Simplifying assumption:
  - Uniform size packets (WLOG, unit size)
  - Unit rates (i.e., $r_Q = r = 1$)

Resource Augmentation is Required

Theorem

Any online algorithm $\textsc{Alg}$ *using* $B_Q \leq B$ *cannot be competitive*

Proof.

- Assume $B_Q = B$.
- $t = 0$ arrival: $\boxed{\text{Y}}$,$\boxed{\text{G}}$
- $t = 1$ forwarding: two cases
  - $\boxed{\text{Y}}$ dropped: $\textsc{Alg}$ cannot be competitive
    - since no more yellow packets may arrive
  - $\boxed{\text{Y}}$ forwarded, and $\boxed{\text{G}}$ enqueued
    - $t = 2$ arrival: burst of B $\boxed{\text{G}}$s
    - $\textsc{Alg}$ has to drop $\boxed{\text{G}}$

Infeasible!!

□

Preliminaries and Basic Concepts

- Use a buffer of size $(1+\varepsilon)B$
  - $\textsc{Opt}$ will use a buffer of size B
- Notation
  - $B_A(t)$: set of packets in the buffer at time t under algorithm A
  - $d_t^A(p)$: buffer position of packet p at time t under algorithm A
- Lower bound on $\textsc{Opt}$ buffer occupancy
  - use a *simulator* $\textsc{Sim}$
  - $\textsc{Sim}$ has the same resources as $\textsc{Opt}$ (rate, buffer)
  - $\textsc{Sim}$ ignores all yellow packets
  - properties of $\textsc{Sim}$:
    - $\forall t$ and $\forall$ green p that arrived by t, $d_t^{\textsc{Sim}}(p) \leq d_t^{\textsc{Opt}}(p)$
    - $\forall t$ $|B_{\textsc{Sim}}(t)| = k$ implies $|B_{\textsc{Opt}}(t)| \geq k$
- A naïve approach:
  - maintain two queues
    - green (size B) and yellow (size εB)
  - always give priority to green queue
    - green queue is equivalent to $\textsc{Sim}$
  - but...

Concept of Lag

- Lag of a green packet at time t under algorithm A:

$$\mathrm{lag}_t^A(p) = d_t^A(p) - d_t^{\mathrm{SIM}}(p)$$

  - for green $p, p' \in B_A(t)$, lag is monotone non-dec. in arrival
  - Context: after forwarding substep or arrival substep?
    - Depends

δ-lag property

Algorithm A satisfies the δ-lag property if $\forall t, p$, $\mathrm{lag}_t^A(p) \leq \delta$

Lag of an algorithm

The lag of algorithm A at time t is $\phi_A(t) = \max_{p \in B_A(t)} \mathrm{lag}_t^A(p)$

- $\phi_A(t)$ determined by last green packet in $B_A(t)$
  - FIFO + monotonicity of lag
- For now: consider lag at end of forwarding substep

Algorithm ON

Algorithm 1 ON: upon the arrival of a new packet p

```
if p is yellow then
    accept if there's room
else                                                  ▷ p is green
    Drop as few yellow packets from the tail of the queue such
    that the new packet will have lag at most εB
    Accept p
end if
```

- omit ON sub/super-scripts: use $\text{lag}_t(p)$, $\phi(t)$

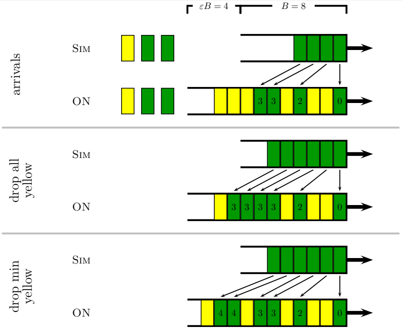
Feasibility of ON

Theorem (Feasibility and lag)

At any time t algorithm ON

- *accepts all green packets*
- *always holds* $\leq (1+\varepsilon)B$ *packets*
- *satisfies the* εB*-lag property*

"Proof by picture"
(formally, by induction)



Competitive Ratio of ON

Theorem (Competitive ratio)

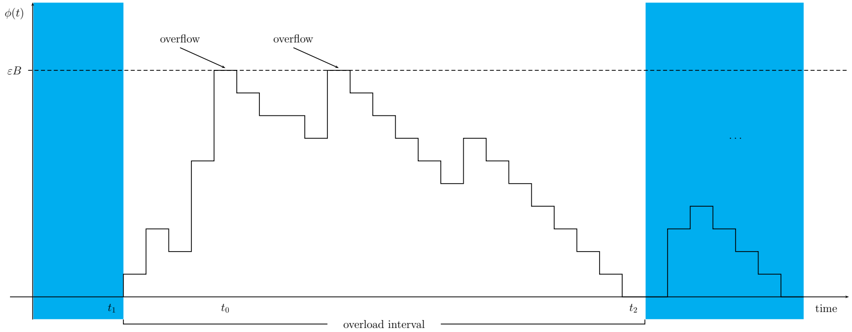
Algorithm ON *is* $\frac{\varepsilon}{1+\varepsilon}$*-competitive*

- Analysis in a nutshell
  - identify "reset" events
  - overflow (yellow dropped) occurs only between resets
    - overflow intervals
  - at least εB yellow are "safe" since last reset
  - many green accepted by $\textsc{Sim}$
    - $\textsc{Opt}$ must deal with them too
    - has little room for many yellow

Notation and Definitions

- From now on: consider lag at end of delivery substep
- Some additional notation and definitions
  - A_t^G: the set of green packets arriving at t
  - A_t^Y: the set of yellow packets arriving at t
  - for simplicity: assume each such set is handled as a batch
    - first green, then yellow
- Safe packets
  - Yellow $p \in B_{\mathrm{ON}}(t)$ *turns safe at $t+1$*
    - t is minimal s.t. $A_t^G \neq \emptyset$, and p is not dropped at t
  - S_t: set of packets turning safe at t
  - for every time interval I, $S(I) = \cup_{t \in I} S_t$
- Reset
  - ON is *reset at t* if $\phi(t) = 0$
    - queue idle at t implies reset at t
    - no green in buffer at t implies reset at t

Overload Intervals: Following the Lag Process



- Between overload intervals, ON does at least as good as OPT
- Suffices to analyze performance in overload intervals
- Any two overload intervals are independent
- Focus on a single overload interval, *I*

Understanding Changes in Lag

Lemma

For any non-reset $t \in I$ s.t. $\phi(t) > 0$,
$\phi(t) = \phi(t-1) + |S_t| - (1 - \mathbb{1}_{\text{SIM}}(t))$

Proof.

- Case 1: $A^G_{t-1} = \emptyset$
  - necessarily $S_t = \emptyset$ (by definition of turning safe)
  - $\phi(t) > 0$:
    - last green $p \in B_{\text{ON}}(t-1)$ still in buffer at the end of t
  - p advances
    - $\mathbb{1}_{\text{SIM}}(t)$ places in $\textsc{Sim}$
    - one place in ON
  - $\text{lag}_t(p) = \text{lag}_{t-1}(p) - (1 - \mathbb{1}_{\text{SIM}})$

$\Rightarrow$ $\phi(t) = \phi(t-1) - (1 - \mathbb{1}_{\text{SIM}}(t)) = \phi(t-1) + |S_t| - (1 - \mathbb{1}_{\text{SIM}}(t))$

Understanding Changes in Lag

Lemma

For any non-reset $t \in I$ s.t. $\phi(t) > 0$,
$\phi(t) = \phi(t-1) + |S_t| - (1 - \mathbb{1}_{\mathrm{SIM}}(t))$

Proof.

- Case 2: $A^G_{t-1} \neq \emptyset$
  - consider
    - last green $p \in B_{\mathrm{ON}}(t-1)$ (end of $t-1$), and
    - last green $p' \in B_{\mathrm{ON}}(t-1)$ (end of delivery substep of $t-1$)
    - p' exists since otherwise reset at $t-1$, and hence in t
  - relative position of p and p' at end of $t-1$
    - p is exactly $\left|A^G_{t-1}\right| + |S_t|$ positions behind p' in ON
    - p is exactly $\left|A^G_{t-1}\right|$ positions behind p' in SIM
  - $\mathrm{lag}_t(p) = \mathrm{lag}_{t-1}(p') + |S_t| - (1 - \mathbb{1}_{\mathrm{SIM}}(t))$
    - p not sent at t since $\phi(t) > 0$

$\Rightarrow$ $\phi(t) = \phi(t-1) + |S_t| - (1 - \mathbb{1}_{\mathrm{SIM}}(t))$ $\square$

Understanding Load on $\textsc{Opt}$ During Overload

Lemma

$\textsc{Sim}$ *delivers* $|I| - |S(I)|$ *green packets during* I

Proof.

- R_I: set of green packets delivered by $\textsc{Sim}$ during I
- $\Delta(t) = \phi(t) - \phi(t-1) = |S_t| - (1 - \mathbb{1}_{\textsc{Sim}}(t))$
- On one hand

$$\begin{aligned} \sum_{t\in I} \Delta(t) &= \sum_{t\in I} [|S_t| - (1 - \mathbb{1}_{\textsc{Sim}}(t))] \\ &= \sum_{t\in I} |S_t| - \sum_{t\in I} 1 + \sum_{t\in I} \mathbb{1}_{\textsc{Sim}}(t) \\ &= |S(I)| - |I| + |R_I| \end{aligned}$$

- On the other hand, $\sum_{t\in I} \Delta(t) = 0$
  - telescopic sum

$\Rightarrow |R_I| = |I| - |S(I)|$ $\square$

Proof of Competitive Ratio

Theorem (Competitive ratio)

Algorithm ON *is* $\frac{\varepsilon}{1+\varepsilon}$*-competitive*

Proof (focus on single overload interval I).

- Recall: OPT has rate $r = 1$ and a buffer of size B
  - OPT deals with at most $r|I| + B = |I| + B$ packets altogether
  - has to deal with at least $|R_I|$ green handled by SIM

$\Rightarrow$ # yellow packets handled by OPT during I is at most
$(|I| + B) - |R_I| = (|I| + B) - (|I| - |S(I)|) = |S(I)| + B$

- $|S(I)| \geq \varepsilon B$
  - definition of overflow interval
  - $\phi(t)$ unit-increase implies a yellow packet turning safe
- Competitive ratio is at least

$$\frac{|S(I)|}{|S(I)|+B} \geq \frac{\varepsilon B}{\varepsilon B+B} = \frac{\varepsilon}{1+\varepsilon}$$

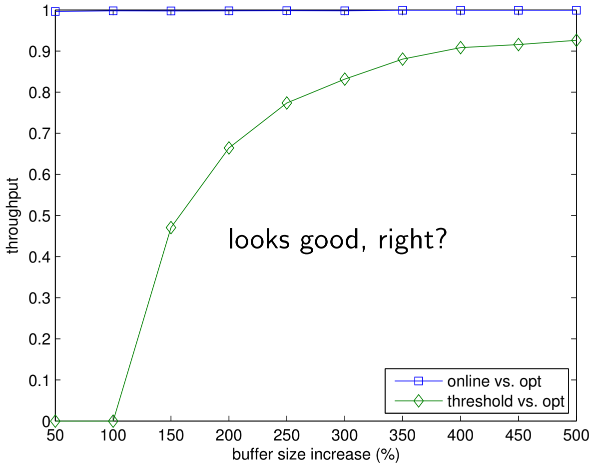
□

Simulation Results

- Traffic
  - bursty MMPP traffic
    - (two) color marked using dual token bucket
  - best-effort Poisson traffic (cross-traffic)
    - zero-rate committed
  - interleaved
- Contending protocols
  - Threshold
    - accept yellow only if buffer occupancy is below threshold T
    - feasibility implies $T = \varepsilon B$ (should be $\leq$)
    - commonly used policy (single-step RED)
  - Naïve protocol
    - 2-queues priority queuing
    - not FIFO, but
    - upper bound on $\textsc{Opt}$
    - serves for normalization

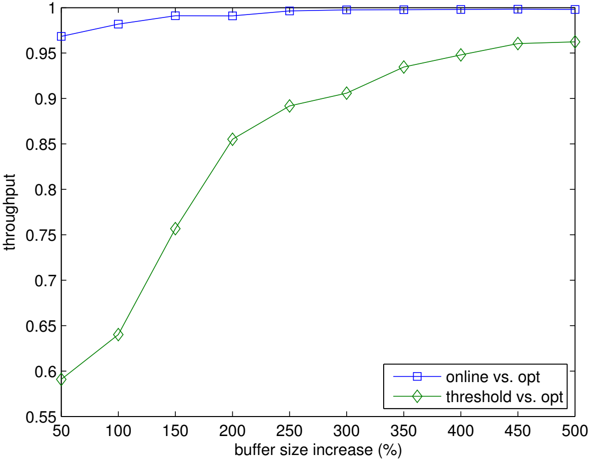
Simulation Results

- Single MMPP aggregate, yellow ∼30%



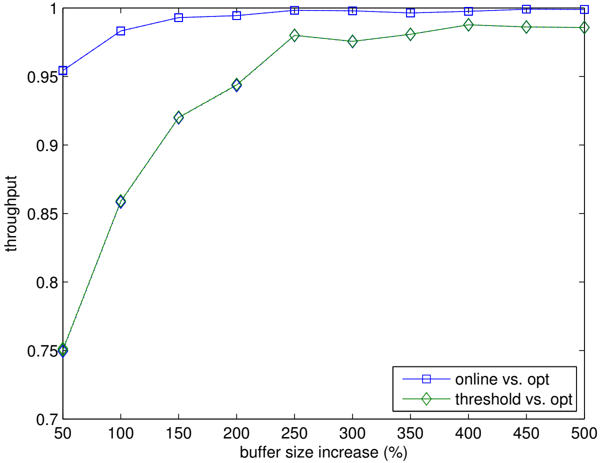
Simulation Results

- MMPP+Poisson, yellow also during OFF, yellow ∼40%



Simulation Results

- MMPP+Poisson, yellow also during OFF, yellow ~50%



Open Questions and Extensions

- Lower bounds
  - for $\varepsilon \leq 1$
    - still with resource augmentation
    - CR is at least ε
- What's the right answer
  - lower bounds for $\varepsilon > 1$?
  - if we double the buffer ($\varepsilon = 1$), do we obtain $\textsc{Opt}$?
  - closing gap for $\varepsilon \in (0, 1]$
- Models that allow dropping "some" committed packets
- Network perspective
  - end-to-end throughput
  - simple topologies (line, tree, . . .)

- References
  - Patt-Shamir, Scalosub and Shavitt, *Competitive Analysis of Buffer Policies with SLA Commitments*, ICNP 2008

A Couple of Lessons Learnt

- FIFO
  - might delay the delivery of "important" packets
  - might have a delayed effect
  - one has to be careful with that
- Competitive approach
  - main focus: find an upper bound on OPT!!!
  - in this analysis: an averaging argument
    - long enough intervals
    - compare ALG and OPT over such intervals
  - later in the course: we'll see other techniques
- Simulation results may be misleading
  - must understand the traffic
  - if it's too good to be true, there's probably a bug...
  - must look at the logs to verify soundness
  - simulations do not "prove" anything
    - traffic might be too "easy" / "hard"
    - difficult to represent real life
    - usually serves to further validate analytic results