Network QoS
371-2-0213

Lecture 9

Gabriel Scalosub

Outline

Simple Packet Scheduling

- FIFO
  - simple to implement
  - sources have incentives to be aggressive
    - increases share of bandwidth
    - "unfair"
  - no mechanism for flow/aggregate protection
  - hard to provide delay guarantees
- Static-Priority
  - provides protection
  - starvation of low-priority
- Weighted Round-Robin
  - predetermined TDM schedule
  - tries to provide $\frac{w_i}{\sum_j w_j}$ service
  - round length depends upon
    - number of flows
    - average packet length
  - may lead to large delays / unfairness
    - e.g., if a packet just misses its slot

Introduction to Generalized Processor Sharing (GPS)

- Fluid multiplexing model
  - infinitesimally small quanta of data
  - bit-level weighted round robin
  - implementation?
    - impossible
    - for demultimplexing, we need "headers"
  - used as a reference model
- Weights:
  - enable unequal share of bandwidth
  - enable traffic differentiation
- Session:
  - aggregate of flows
  - share same queue in the round-robin schedule
  - DiffServ:
    - correspond to same DSCP/PHB
  - in general
    - may depend on many parameters (e.g., 5-tuple)

GPS Model and Architecture

- Work conserving policy, with rate r
- Separate FIFO queue for each session
- Assume N queues/sessions
  - ϕ_i: weight of queue i ($i = 1, \ldots, N$)
  - $S_i(\tau, t)$: amount of session i's flow delivered during $(\tau, t]$.
- GPS scheduler:
  - for any continuously backlogged session i during $(\tau, t]$

$$\frac{S_i(\tau, t)}{S_j(\tau, t)} \geq \frac{\phi_i}{\phi_j} \quad j = 1, \ldots, N$$

  - in particular, summing over all j

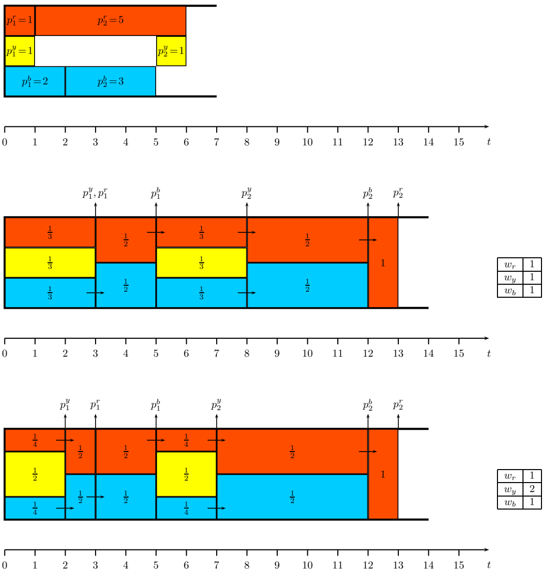
$$S_i(\tau, t) \sum_j \phi_j \geq \phi_i \sum_j S_j(\tau, t) = \phi_i (t - \tau) r$$

    - i backlogged + work conserving: $\sum_j S_j(\tau, t) = (t - \tau) r$
    - implies i's rate is at least $g_i = \frac{\phi_i}{\sum_j \phi_j} r$

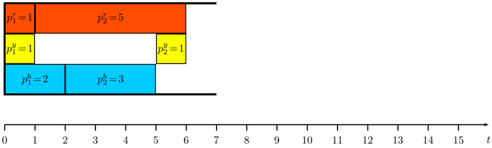
GPS Preliminary Properties

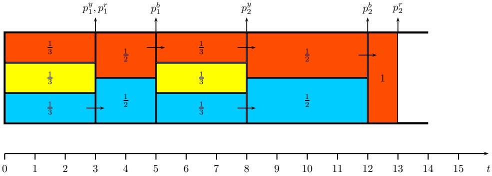
- Bandwidth
  - r_i: session i's average rate
  - if $r_i \leq g_i$, average rate r_i can be guaranteed
  - backlog is cleared at rate $\geq g_i$
- Delay
  - session i's delay depends only on i's queue length
- Sensitivity to weights
  - if $\sum_i r_i < r$, any weight-assignment yields a stable system
    - stability: probability of unbounded buffers is 0
  - weight-assignment captures delay-tolerance
  - e.g., high-BW delay-insensitive session i can settle for $g_i < r_i$

GPS Example: Fluid-Pipe Diagram

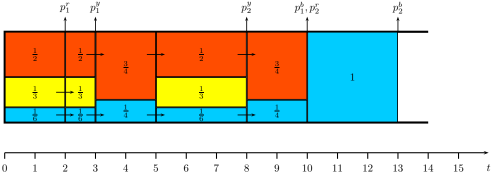


GPS Example: Fluid-Pipe Diagram



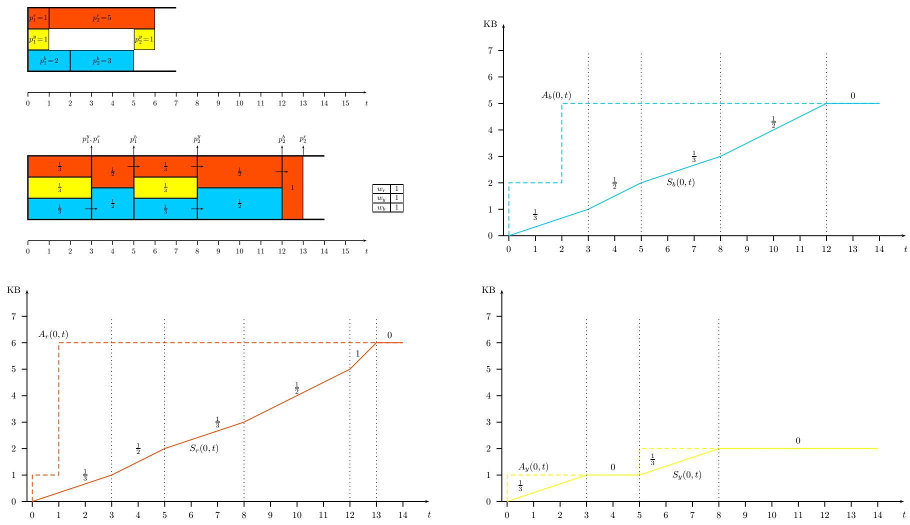


w_r	1
w_y	1
w_b	1



w_r	3
w_y	2
w_b	1

GPS Example: Service Curves



Ideals vs. Reality

- GPS:
  - fluid model
  - multiple queues serviced “simultaneously”
  - optimal bandwidth sharing
    - max-min weighted fairness
- Real life:
  - packetized traffic
    - do not preempt a packet being sent
  - one queue can be served at any given time
  - can we be as good as GPS in spite of packetization?
- Bridging ideals and reality
  - need to approximate fluid-based GPS by a packetized algorithm
- Solution
  - Packet-by-packet GPS (PGPS)
    - Parekh and Gallager (1993)
  - a.k.a. Weighted Fair Queuing (WFQ)
    - Demers, Keshav and Shenker (1990)

How to Packetize GPS?

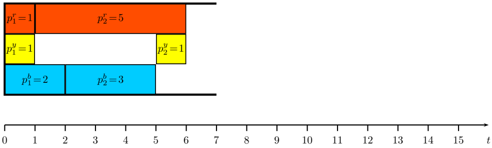
- Offline:
  - assume all packets arrive at $t = 0$
  - can we rearrange the schedule into a packetized one?
  - can we do this s.t. packets finish no-later than in GPS?
  - yes!! (on both accounts)
    - similar proof to that of GIS for interval scheduling (last week)
- Online basic idea:
  - follow GPS as closely as possible
  - simulate GPS to produce virtual deadlines
  - employ EDF
  - recall: care about bandwidth sharing, not making the deadlines

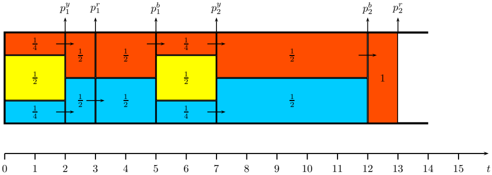
Algorithm 1 PGPS: at any idle-link time t

1: **for** every pending packet p **do**
2: $F_p \leftarrow p$'s departure time in GPS ▷ simulate GPS
3: **end for**
4: forward $p' = \arg\min \{F_p \mid \text{pending } p\}$

Is PGPS As Good As GPS?

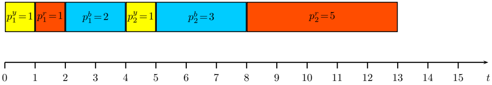
- Consider





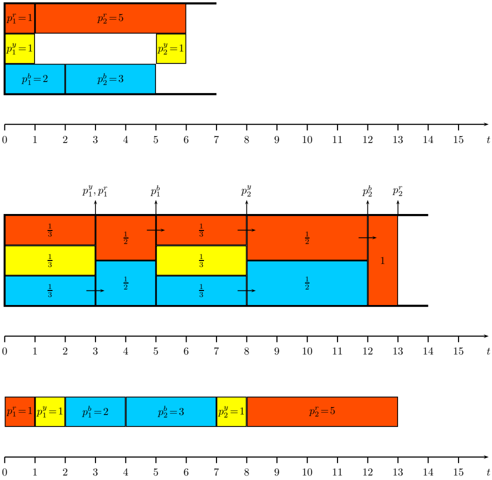
w_r	1
w_y	2
w_b	1

- The (order-maintaining) packetized version would be



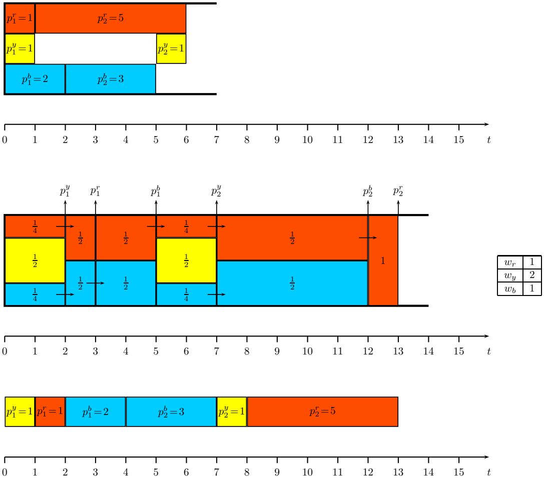
- Is this feasible?
- Delay vs. work-conserving

GPS vs. PGPS

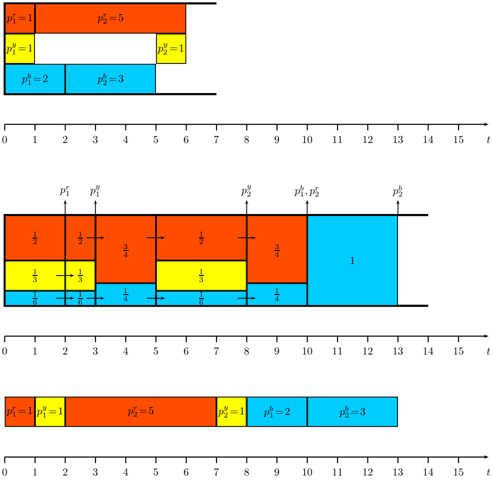


w_r	1
w_y	1
w_b	1

GPS vs. PGPS


$p_1^r = 1$
$p_2^r = 5$
$p_1^y = 1$
$p_2^y = 1$
$p_1^b = 2$
$p_2^b = 3$
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 t
p_1^y p_1^r p_1^b p_2^y p_2^b p_2^r
$\frac{1}{4}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{2}$ 1
$\frac{1}{2}$ $\frac{1}{2}$
$\frac{1}{4}$ $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{2}$
w_r 1
w_y 2
w_b 1

GPS vs. PGPS



w_r	3
w_y	2
w_b	1

Order Property

- In PGPS
  - F_p is only calculated w.r.t. packets already arrived
  - why is this consistent with "offline" GPS?

Lemma

Let p, p' be packets in a GPS system at time t.
If p is completed before p' with no arrivals after t,
then the same holds for any arrival pattern after t.

Proof (almost shorter than claim...).

- The sessions of p, p' are both backlogged until we complete the first of the two
- The service ratio between these sessions is independent of future arrivals: $\phi_i / \phi_{i'}$
- Order is maintained

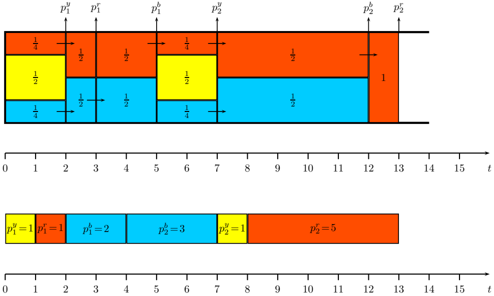
□

Lag Property

- Notation:
  - F_p: p's departure time in GPS
  - $\hat{F}_p$: p's departure time in PGPS
  - $L_{\max}$: maximum packet length
  - r: outgoing link rate

Corollary

If $\hat{F}_p > F_p$ then p arrived too late to be transmitted in GPS order



w_r	1
w_y	2
w_b	1

Lag Property

Theorem (how late can PGPS be)

$\hat{F}_p - F_p \leq \frac{L_{\max}}{r}$

Proof.

- GPS and PGPS are work conserving
  - busy periods coincide
  - suffices to prove per busy period
- Notation:
  - p_k: k'th packet to depart under PGPS
  - L_k: p_k's length
  - a_k: p_k's arrival time
  - t_k: p_k's departure time under PGPS
  - u_k: p_k's departure time under GPS
- We'll show (for all k):
  - $t_k \leq u_k + \frac{L_{\max}}{r}$

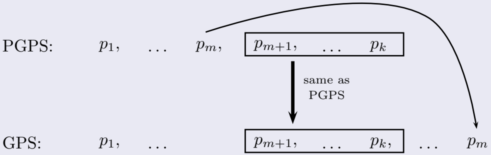
Lag Property

Theorem (how late can PGPS be)

$$\hat{F}_p - F_p \leq \frac{L_{\max}}{r}$$

Proof.

- Case 1: $p_1, \ldots, p_{k-1}$ depart before p_k under GPS
  - busy-period: GPS had to finish work on $p_1, \ldots, p_{k-1}$
  - $t_k \leq u_k$
- Case 2: exists $0 < m \leq k-1$
  - p_m departs after p_k under GPS, i.e. $u_m > u_k$
  - m is maximal
  - i.e., $p_{m+1}, \ldots, p_{k-1}$ depart no later than p_k under GPS



Lag Property

Theorem (how late can PGPS be)

$\hat{F}_p - F_p \leq \frac{L_{\max}}{r}$

Proof.

- p_m begins tranmission at $t_m - \frac{L_m}{r}$ under PGPS
- By lemma:

$$\min\{a_{m+1}, \ldots, a_k\} > t_m - \frac{L_m}{r}$$

  - Otherwise
    - they all depart no later than p_k under GPS
    - PGPS would have sent one of them
- $p_{m+1}, \ldots, p_{k-1}$
  - arrive after $t_m - \frac{L_m}{r}$
  - depart no later than p_k under GPS
- Hence

$$u_k \geq \frac{1}{r}(L_{m+1} + \ldots + L_k) + t_m - \frac{L_m}{r} \geq t_k - \frac{L_{\max}}{r} \qquad \square$$

Backlog Property

Theorem (how backlogged can PGPS be compared to GPS)

For all i, t, $S_i(0,t) - \hat{S}_i(0,t) \leq L_{\max}$

Proof.

- Comparing slopes
  - $\hat{S}_i$
    - r: i's packet is being transmitted
    - 0: otherwise
  - S_i
    - always in $[0, r]$

$\Rightarrow$ Maximal difference: i's packets begin transmission in PGPS

- Consider i's packet p
  - t: p begins transmission in PGPS (length L)
  - completed at $t + \frac{L}{r}$
  - τ: completed by GPS

Backlog Property

Theorem (how backlogged can PGPS be compared to GPS)

For all i, t, $S_i(0,t) - \hat{S}_i(0,t) \le L_{\max}$

Proof.

- FIFO within i (both GPS and PGPS)
  - $S_i(0,\tau) = \hat{S}_i(0, t + \frac{L}{r})$
- By previous theorem
  - $\tau \ge (t + \frac{L}{r}) - \frac{L_{\max}}{r}$
- Hence
  $S_i(0, t + \frac{L - L_{\max}}{r}) \le S_i(0,\tau) = \hat{S}_i(0, t + \frac{L}{r}) = \hat{S}_i(0,t) + L$
- S_i's slope $\le r$:

$$\begin{aligned} S_i(0,t) &= S_i(0, t + \tfrac{L-L_{\max}}{r}) + S_i(t + \tfrac{L-L_{\max}}{r}, t) \\ &\le \hat{S}_i(0,t) + L + r \cdot \tfrac{L_{\max} - L}{r} \\ &= \hat{S}_i(0,t) + L_{\max} \end{aligned}$$

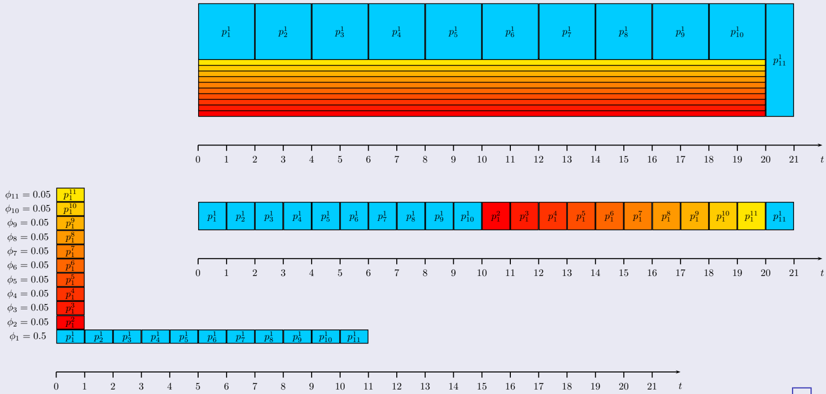
$\square$

How Far Ahead Can PGPS Be Compared To GPS?

Lemma

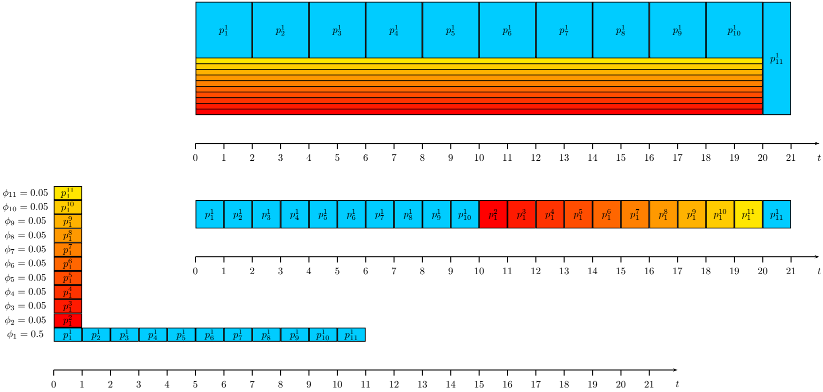
There is no constant $c > 0$ s.t. $\forall i, t$, $\hat{S}_i(0,t) - S_i(0,t) \leq c \cdot L_{\max}$

Proof (by picture).



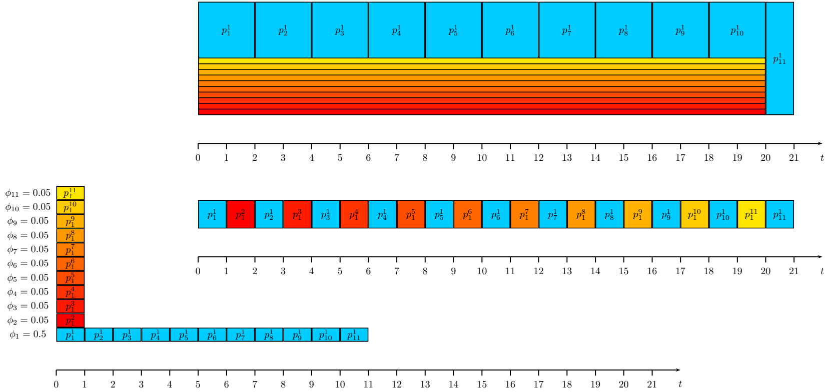
Worst Case Fair WFQ (WF^{2}Q)

- Another packetized approximation of GPS
- WFQ vs. WF^{2}Q
  - WFQ: consider only GPS finish times
    - $\arg\min \{F_p \mid \text{pending } p\}$
  - WF^{2}Q: consider both GPS start and finish times
    - $\arg\min \{F_p \mid \text{pending } p \text{ already in service under GPS}\}$



Worst Case Fair WFQ (WF^{2}Q)

- Another packetized approximation of GPS
- WFQ vs. WF^{2}Q
  - WFQ: consider only GPS finish times
    - $\arg\min \{F_p \mid \text{pending } p\}$
  - WF^{2}Q: consider both GPS start and finish times
    - $\arg\min \{F_p \mid \text{pending } p \text{ already in service under GPS}\}$



Observations and Other Algorithms

- WFQ/WF^{2}Q:
  - E2E delay guarantees for leaky-bucket sources
  - recall IntServ...

- Self-Clocked Fair Queuing (SCFQ)
  - does not simulate GPS: "local" approximation of F_p
  - depend on current serviced packet + previous session's packet
  - simpler implementation than WFQ/WF^{2}Q
- Deficit Round-Robin (DRR)
  - credit-based WRR
  - schedule packet in time slot only if sufficient credit
  - time slot size independent of packet lengths
  - simpler implementation than WFQ/WF^{2}Q
- Non work-conserving
  - Stop-and-go
    - delay by fixed-window length before sending
    - can be used to regulate jitter
- Many others...

References

- Parekh and Gallager. A generalized processor sharing approach to flow control in integrated services networks: The Single Node Case. IEEE/ACM Transactions on Networking 1(3): 344–357 (1993)
- Demers, Keshav and Shenker. Analysis and simulation of a fair queueing algorithm. Internetworking: Research and Experience 1: 3–26 (1990)
- Zhang. Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks. Proceedings of the IEEE 83(10): 1374–1396 (1995)