Aalto University
School of Science
and Technology

Flow-level stability and performance of channel-aware priority-based schedulers

Samuli Aalto

Pasi Lassila

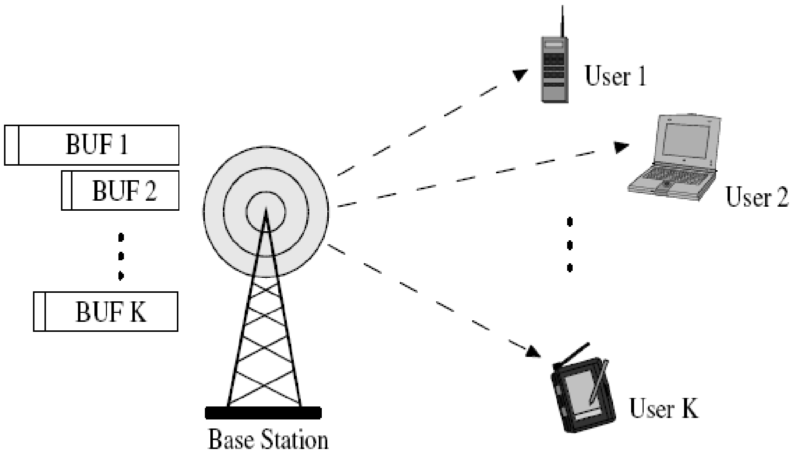
NGI 2010
2-4 June 2010
Paris, France

Outline

- Problem formulation
- Channel-aware priority-based schedulers
- Stability results
- Numerical study
- Conclusions

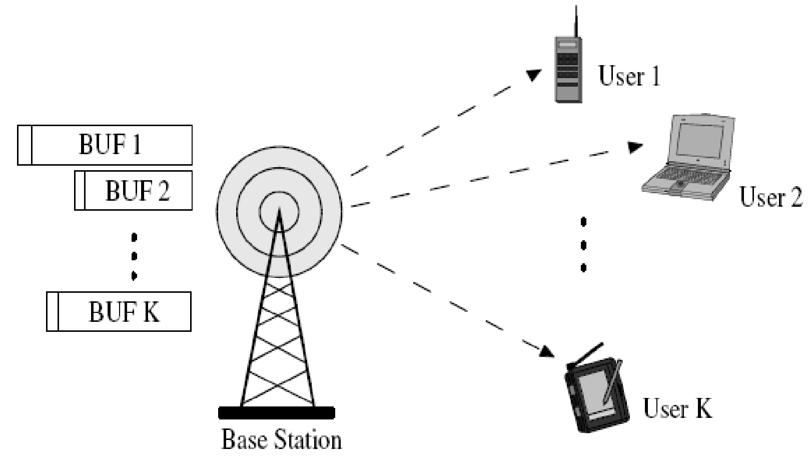
Introduction

- Downlink data transmission in a cellular system
- Traffic consists of elastic flows
  - file transfers using TCP
- Base station transmits to a single user in a time slot
  - decided by the scheduler
  - time scale of milliseconds
- Dynamic traffic setting
  - random arrivals and departures of users (= flows)
  - time scale of seconds
- Flow-level stability and perform-ance of various schedulers?



Schedulers

- Non-channel-aware schedulers
  - Scheduling based on average rate information
  - Example: Round Robin (RR) scheduler
- Channel-aware schedulers
  - Scheduling based on instantaneous rate information
  - Examples: Maximum Rate (MR) scheduler, Relative Best (RB) scheduler, Proportional Fair (PF) scheduler



User model

- K user classes
- Class-k users have stationary IID rate processes $R_i(t)$
  - Mean rate r_k (bps)
  - Maximum rate r_k* (bps)
- If user i scheduled at time slot t, the corresponding flow is served with rate $R_i(t)$

Channel-aware scheduling

- Base station
  - knows the instantaneous rates $R_i(t)$ of all active users i
  - can favor those users having instantaneously good channel
- Static setting
  - Queue length-based policies shown to have many desirable properties [Mandelbaum and Stolyar (2004), Stolyar (2005)]
- Not much work on dynamic setting
  - Seminal work on stability by [Borst (2005), Borst and Jonckheere (2006)]
  - Minimizing mean delay very difficult and hardly anything is known

Utility-based schedulers

- Base station knows
  - instantaneous rates $R_i(t)$ of all active users i
  - throughputs $T_i(t)$ of all active users i
- Definition: Scheduling based on
  - utility function $U(\theta)$
  - time slot t allocated to user $i*$ such that
    - $i* = \arg\max_i R_i(t)\, U'(T_i(t))$
- Examples:
  - Alpha-fair schedulers
  - Proportional Fair (PF) scheduler

Alpha-fair schedulers

- Definition: Utility-based scheduler with utility function
  - $U(\theta; \alpha) = (1-\alpha)^{-1} \theta^{1-\alpha}$ $\quad (\alpha \neq 1)$
  - $U(\theta; 1) = \log \theta$ $\quad (\alpha = 1)$
  - originally defined in [Mo and Walrand (2000)]

- Example: Proportional Fair (PF) scheduler
  - alpha-fair scheduler with $\alpha = 1$
  - time slot t allocated to user i^* such that
    - $i^* = \arg\max_i R_i(t)/T_i(t)$
  - implemented in the HDR system [Viswanath et al. (2002)]

Dynamic traffic setting

- Class-k users (= flows) arrive
  - according to an independent Poisson process
  - with rate λ_k (flows per second)
- Flow sizes X_i IID
  - with mean x (bits)
- Flow i departs
  - as soon as all X_i bits of the flow have been transmitted

Traffic load

- Class-k bit arrival rate $\lambda_k x$ (bps)
- Traffic load $\rho_k^* = \lambda_k x / r_k^*$ (w.r.t. the maximum rate)
- Traffic load $\rho_k = \lambda_k x / r_k$ (w.r.t. the mean rate)
- Note: $\rho_k^* < \rho_k$

Known stability results

- Definition: Flow-level stability
  - The total number of flows does not explode!
- Necessary stability condition for non-channel-aware schedulers:
  - $\rho_1 + \ldots + \rho_K \leq 1$ [classic queueing theory]
- Necessary stability condition for channel-aware schedulers:
  - $\rho_1^* + \ldots + \rho_K^* \leq 1$ [Borst and Jonckheere (2006)]
- Sufficient stability condition for alpha-fair schedulers:
  - $\rho_1^* + \ldots + \rho_K^* < 1$ [Borst and Jonckheere (2006)]

Overview

- We study priority-based channel-aware schedulers
  - Priority can be any strictly increasing function of instantaneous rate
  - Includes as special cases many known channel-aware schedulers
- Stability
  - Achieving maximum stability region is a robustness property
  - We give a general condition when the necessary condition is also sufficient
  - When the necessary condition is not sufficient, we give a sufficient condition for some special cases
- Performance
  - We have also made simulation studies to gain insight on actual performance (including comparisons against alpha-fair policies)

Outline

- Problem formulation
- Channel-aware priority-based schedulers
- Stability results
- Numerical study
- Conclusions

Rate-based priority schedulers

- Base station knows
  - instantaneous rates $R_i(t)$ of all active users i
- Definition: Scheduling based on
  - class-specific increasing priority function $h_k(r)$
  - determines the instantaneous priority $P_i(t) = h_{k(i)}(R_i(t))$ of user i
  - time slot t allocated to user $i*$ such that
    - $i* = \arg\max_i P_i(t) = \arg\max_i h_{k(i)}(R_i(t))$
- Examples:
  - Weight-based priority schedulers [Borst (2005)]
  - CDF-based priority schedulers [Park et al. (2005)]

Weight-based priority schedulers (1)

- Definition: Rate-based priority scheduler with
  - linear priority functions $h_k(r) = w_k r$
- Examples:
  - Absolute rate priority schedulers (e.g. MR)
  - Relative rate priority schedulers (e.g. RB)
  - Proportional rate priority schedulers (e.g. PB)
  - MR, RB, and PB break ties within any priority class at random

Weight-based priority schedulers (2)

- Definition: Absolute rate priority scheduler ($w_k = 1$):
  - time slot t allocated to user $i*$ such that
    - $i* = \arg\max_i R_i(t)$
- Definition: Relative rate priority scheduler ($w_k = 1/r_k$):
  - time slot t allocated to user $i*$ such that
    - $i* = \arg\max_i R_i(t)/r_{k(i)}$
- Def: Proportional rate priority scheduler ($w_k = 1/r_k*$):
  - time slot t allocated to user $i*$ such that
    - $i* = \arg\max_i R_i(t)/r_{k(i)}*$

CDF-based priority schedulers

- Definition: Rate-based priority scheduler with
  - non-linear priority functions $h_k(r) = F_k(r)$
  - where $F_k(r) = P\{R_k \le r\}$ is the stationary CDF of the corresponding rate process
- Example: CS breaks ties within any priority class at random

Outline

- Problem formulation
- Channel-aware priority-based schedulers
- Stability results
- Numerical study
- Conclusions

Rate model (1)

- Assumption: a finite number of possible rate values
- Class-k users
  - Maximum rate $r_k{}^*$
  - Second highest rate $r_k{}^{**}$
  - Maximum priority $p_k{}^* = h_k(r_k{}^*)$
  - Second highest priority $p_k{}^{**} = h_k(r_k{}^{**})$
- Note:
  - Proportional rate priority scheduler: $p_k{}^* = 1$ for all k
  - CDF-based priority scheduler: $p_k{}^* = 1$ for all k

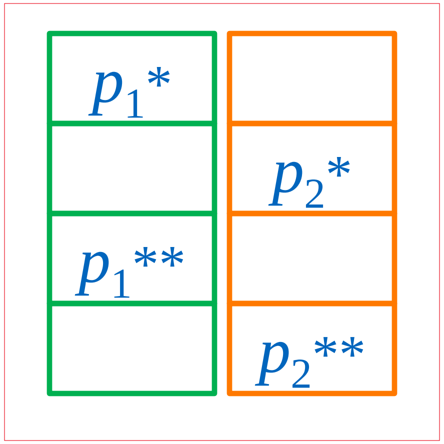
Rate model (2)

- Example: Possible rate values for the HDR system

Index	Rate (kbit/s)
r_1	38.4
r_2	76.8
r_3	102.6
r_4	153.6
r_5	204.8
r_6	307.2
r_7	614.4
r_8	921.6
r_9	1228.8
r_{10}	1843.2
r_{11}	2457.6

Main result

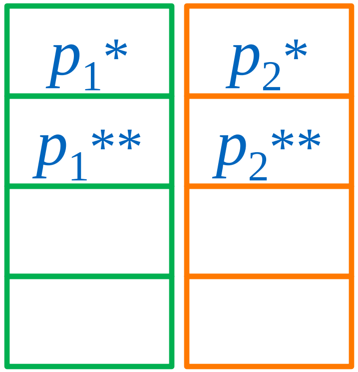
- Consider a rate-based priority scheduler π
- Result 1: If $p_k^* > p_l^{**}$ for all $k \neq l$, then scheduler π is stable under condition
  - $\rho_1^* + \ldots + \rho_K^* < 1$
  - i.e. the same condition as for the alpha-fair schedulers



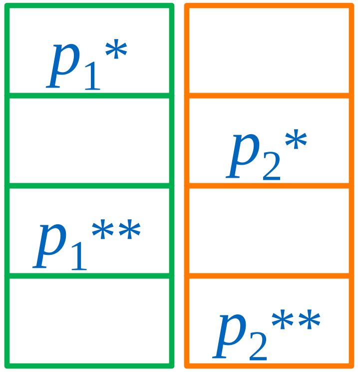
- Intuitive proof:
  - Since $p_k^* > p_l^{**}$ for all $k \neq l$, all classes k will be served with their own maximum rate r_k^* at the stability limit

Corollaries (1)

- If $r_k^* = r_l^*$ for all $k = l$, then any absolute rate priority scheduler is stable under the given condition

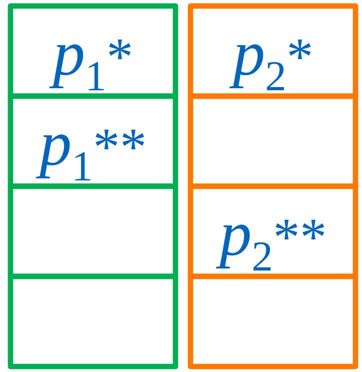


- If $r_k^*/r_k > r_l^{**}/r_l$ for all $k \neq l$, then any relative rate priority scheduler is stable under the given condition

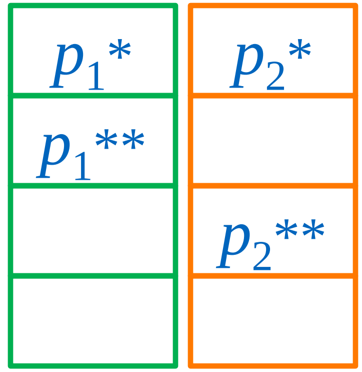


Corollaries (2)

- Any proportional rate priority scheduler is stable under the given condition

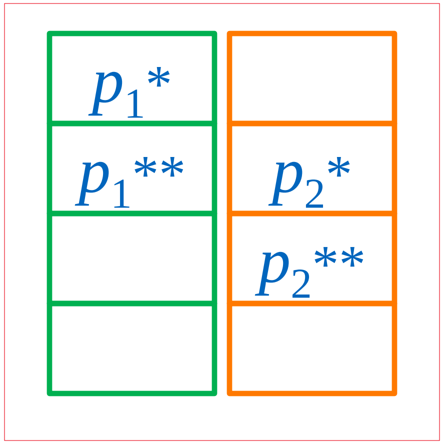


- Any CDF-based priority scheduler is stable under the given condition



Another result

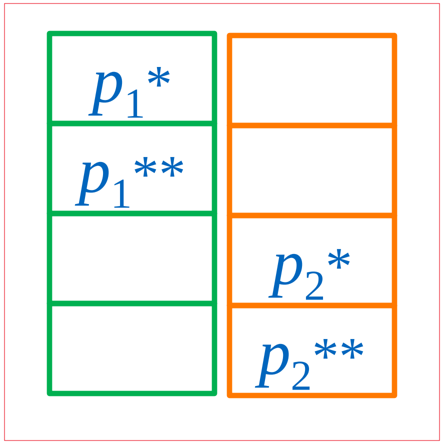
- Consider a rate-based priority scheduler π that breaks ties within any priority class at random
- Result 2: If $p_k^* \geq p_l^{**}$ for all $k \neq l$, then scheduler π is stable under condition
  - $\rho_1^* + \ldots + \rho_K^* < 1$



- Intuitive proof:
  - If $p_k^* = p_l^{**}$ for some $k \neq l$, the tie-breaking rule guarantees that class k will take over class l at the stability limit, and, thus, will be served with its own maximum rate r_k^*

Further stability conditions for K = 2

- Assumption: $K = 2$
- Consider a rate-based priority scheduler π that breaks ties within any priority class at random
- Result 3: If $p_2^* < p_1^{**}$, then scheduler π is stable under condition
  - $P\{h_1(M_1) > p_2^*\} + \rho_2^* < 1$



- Note: The condition above is more stringent than
  - $\rho_1^* + \rho_2^* < 1$

Numerical example

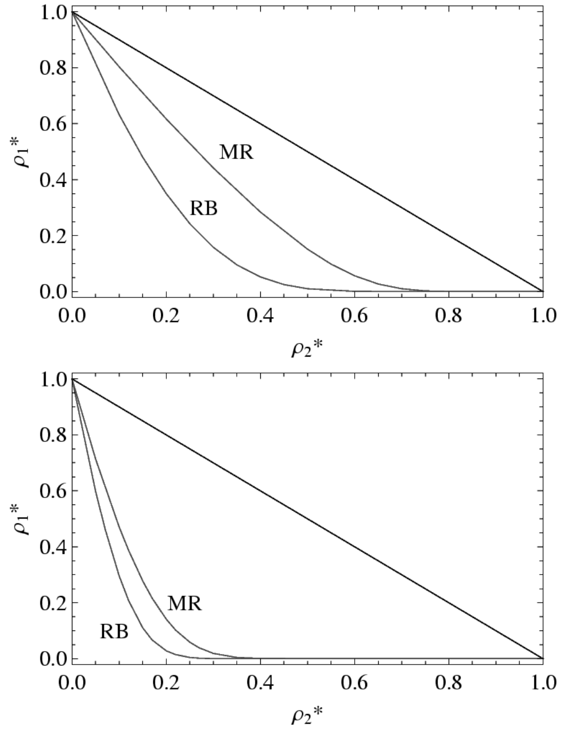


Figure 1: Stability region as a function of ρ_1^* and ρ_2^* when $j_1 = 7$ (upper panel) and $j_1 = 4$ (lower panel).

Impact of a continuous rate distribution

- Consider the case where the rate distribution is continuous, however, with a bounded support
- Is the natural condition below sufficient for stability?
  - $\rho_1^* + \ldots + \rho_K^* < 1$
- Proportional rate and CDF-based priority schedulers are still stable under the given condition
- Absolute rate priority scheduler is stable if the class-specific rate distributions have the same support
- Relative rate priority scheduler needs a more stringent condition for stability

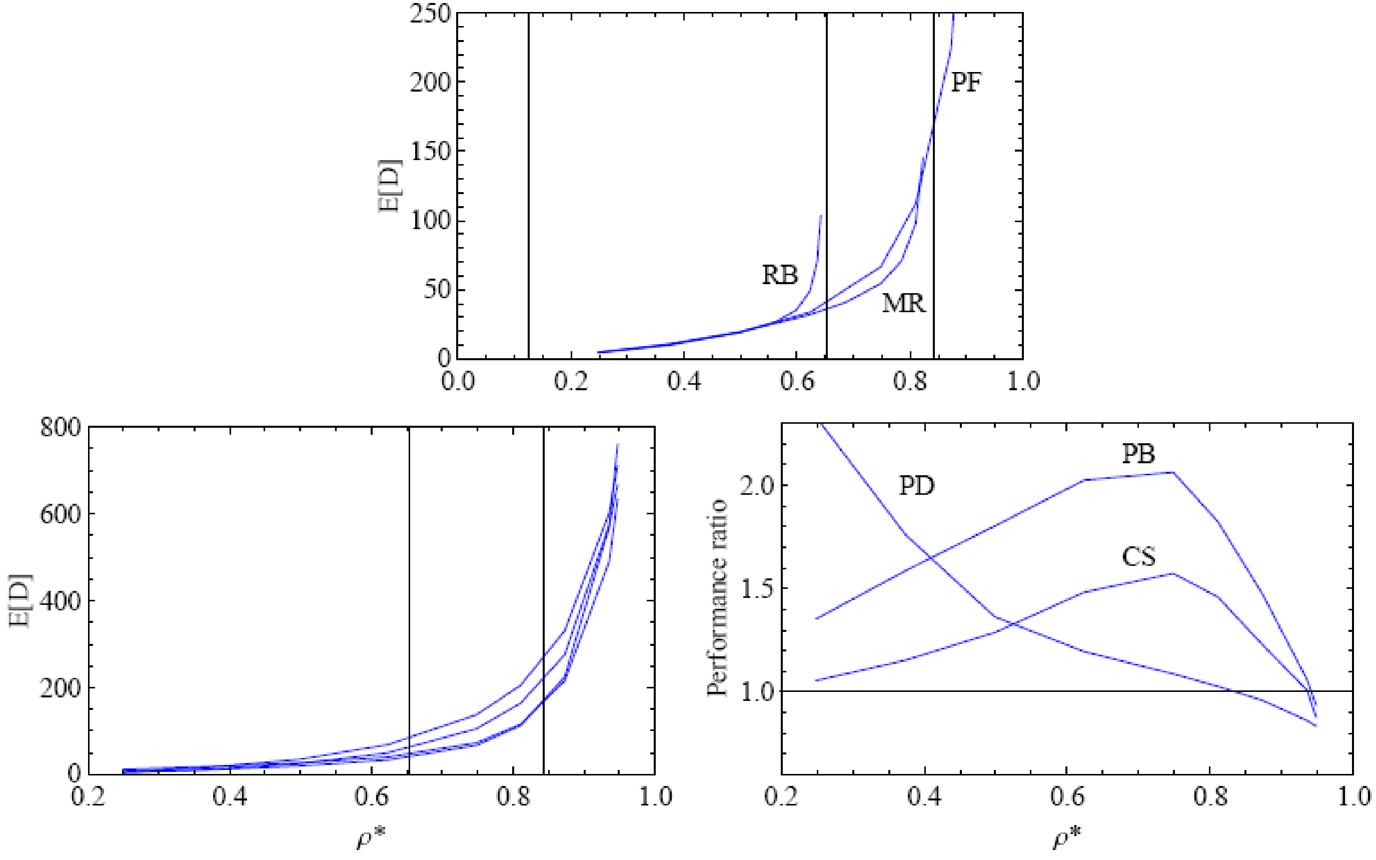
Outline

- Problem formulation
- Channel-aware priority-based schedulers
- Stability results
- Numerical study
- Conclusions

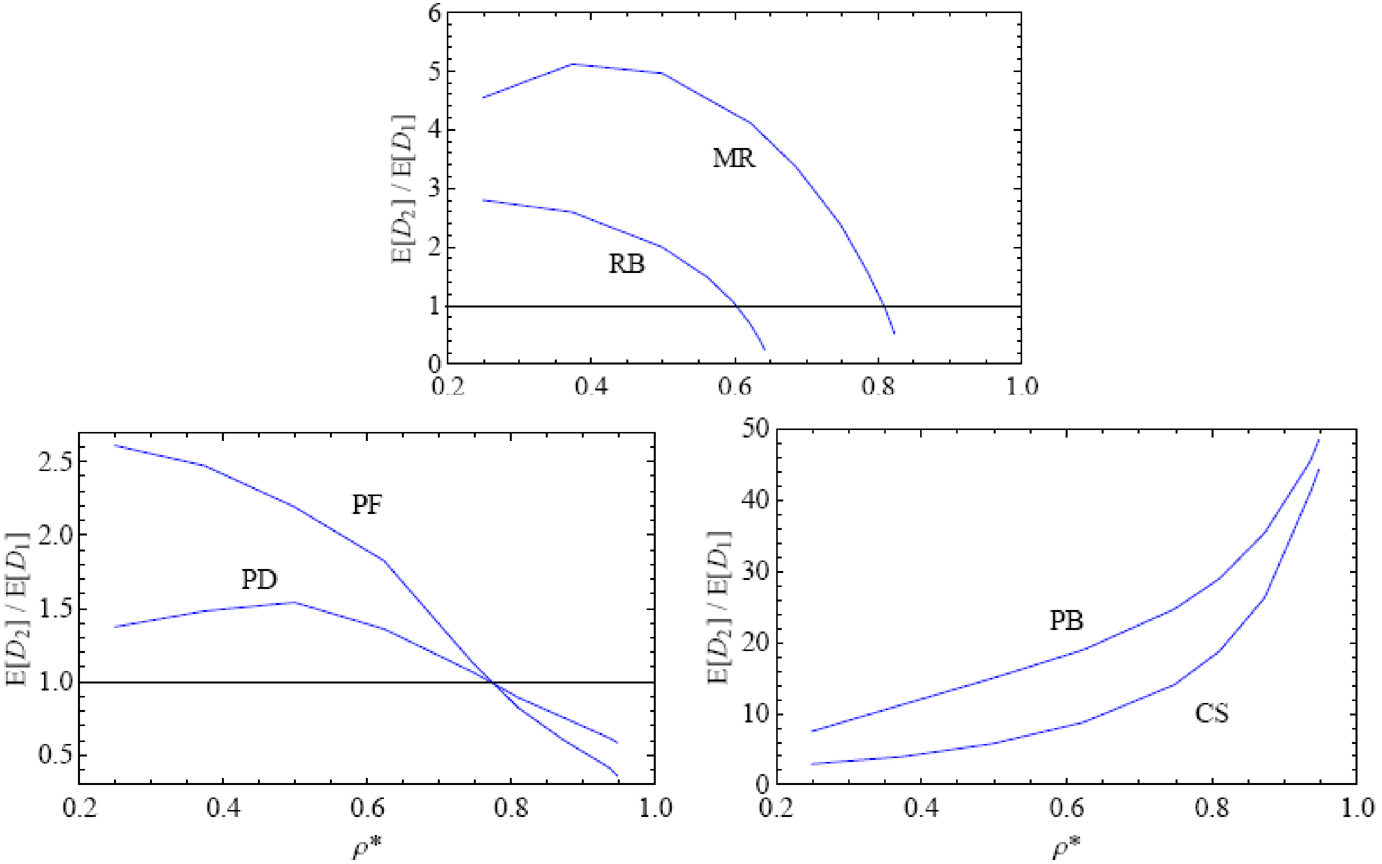
Numerical study

- Schedulers
  - Priority-based: MR, RB, PB and CS with random tie-breaking
  - Utility-based: PF ($\alpha = 1$) and PD ($\alpha = 2$)
  - Also, investigate impact of using SRPT-like tie-breaking rules
- Parameters
  - 2 user classes with flow arrival rates $\lambda_1 = \lambda_2 = 1/2$
  - HDR transmission rates, i.e., 11 possible rate values
    - Class 1 flows can achieve 7 lowest rates
    - Class 2 flows can achieve all 11 rates
  - Truncated geometric rate distributions with parameters $q_1 = 1$, $q_2 = 1/2$

Overall performance (mean delay)

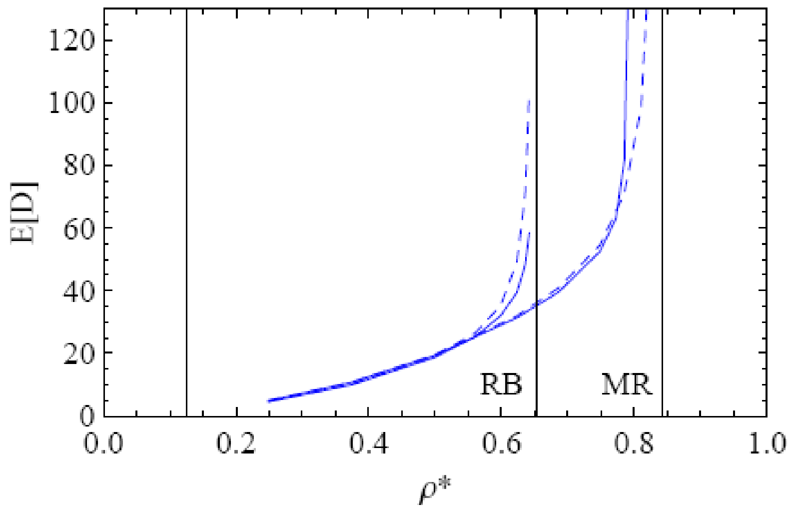


Fairness (mean delay ratio)

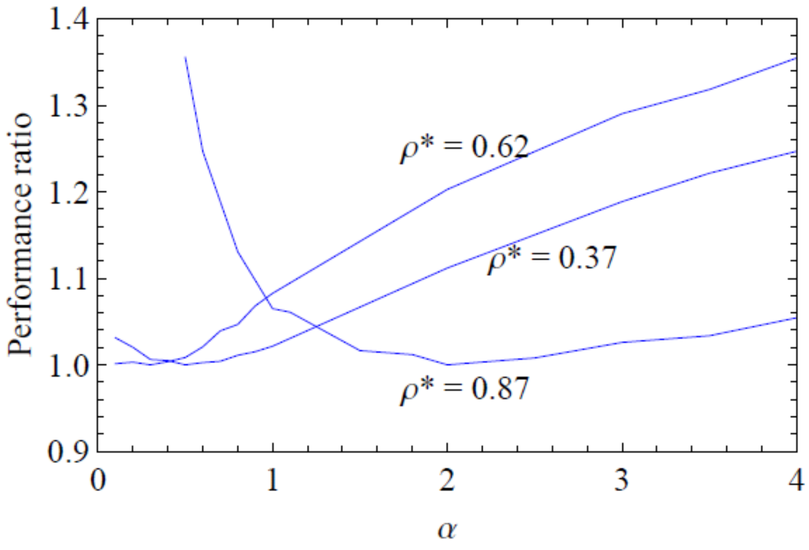


Other performance comparisons

Impact of SRPT-like tie-breaking



Optimizing parameter α



Outline

- Problem formulation
- Channel-aware priority-based schedulers
- Stability results
- Numerical study
- Conclusions

Conclusions

- Stability
  - Proportional rate and CDF-based priority schedulers do have the maximum stability region
  - Absolute rate priority schedulers don't necessarily have
  - Relative rate priority schedulers don't usually have
- Performance
  - MR and RB offer quite good performance but may become unstable
  - PB and CS policies are stable but very unfair
  - PF performs very well over a large region of loads (good overall)
  - PD can outperform PF at very high loads
  - SRPT-like tie-breaking heuristics do not work at the time-slot level
  - To minimize the mean delay, flow-level information can be used to tune the packet level schedulers

The End