Dynamic Windows Scheduling with Reallocation

Martín Farach-Colton[1] Katia Leal[2]
Miguel A. Mosteiro[3] Christopher Thraves[4]

[1]Department of Computer Science, Rutgers University, USA & Tokutek Inc.

[2]GSyC, Universidad Rey Juan Carlos, Madrid, Spain

[3]Dept. of Computer Science, Kean University, Union, NJ, USA

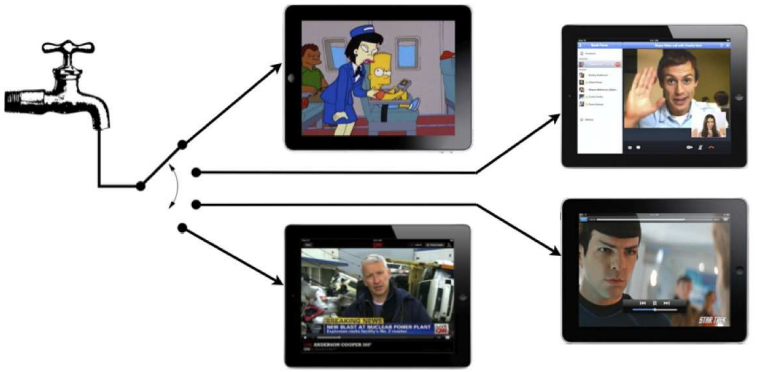
[4]LAAS-CNRS, Toulouse, France

SEA 2014

Motivation

Multiple users need to access a shared limited resource,

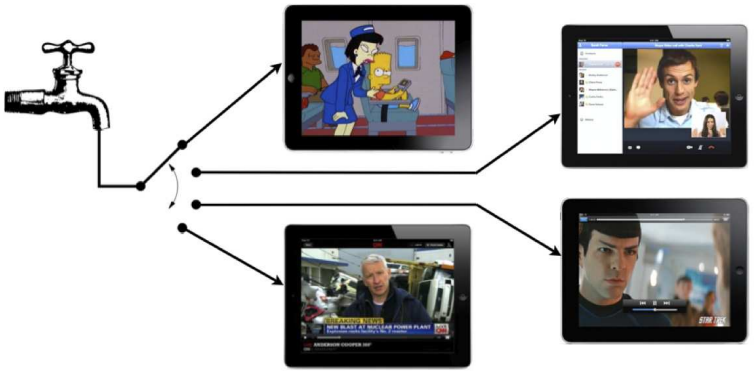
each user can wait for a while ....



... but not too long!

Motivation

- E.g. communication networks, media streaming, inventory replenishment, etc.

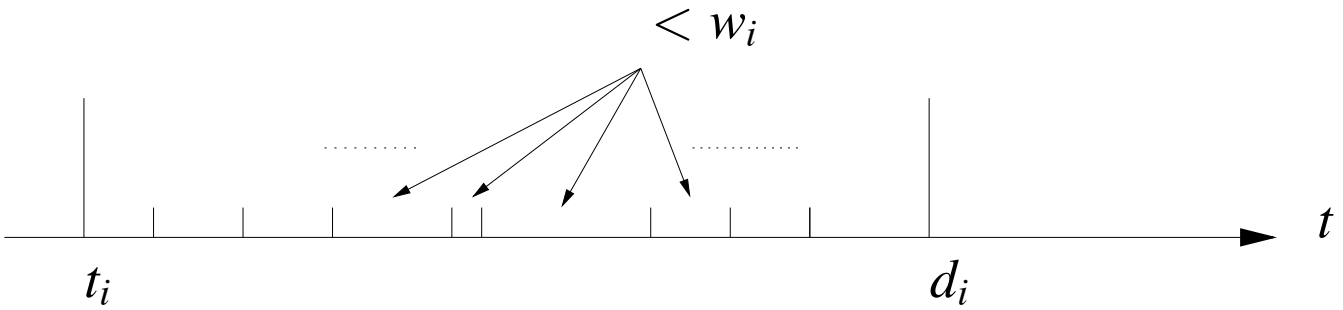


We focus on

scheduling clients' transmissions to communication channels and slotted time.

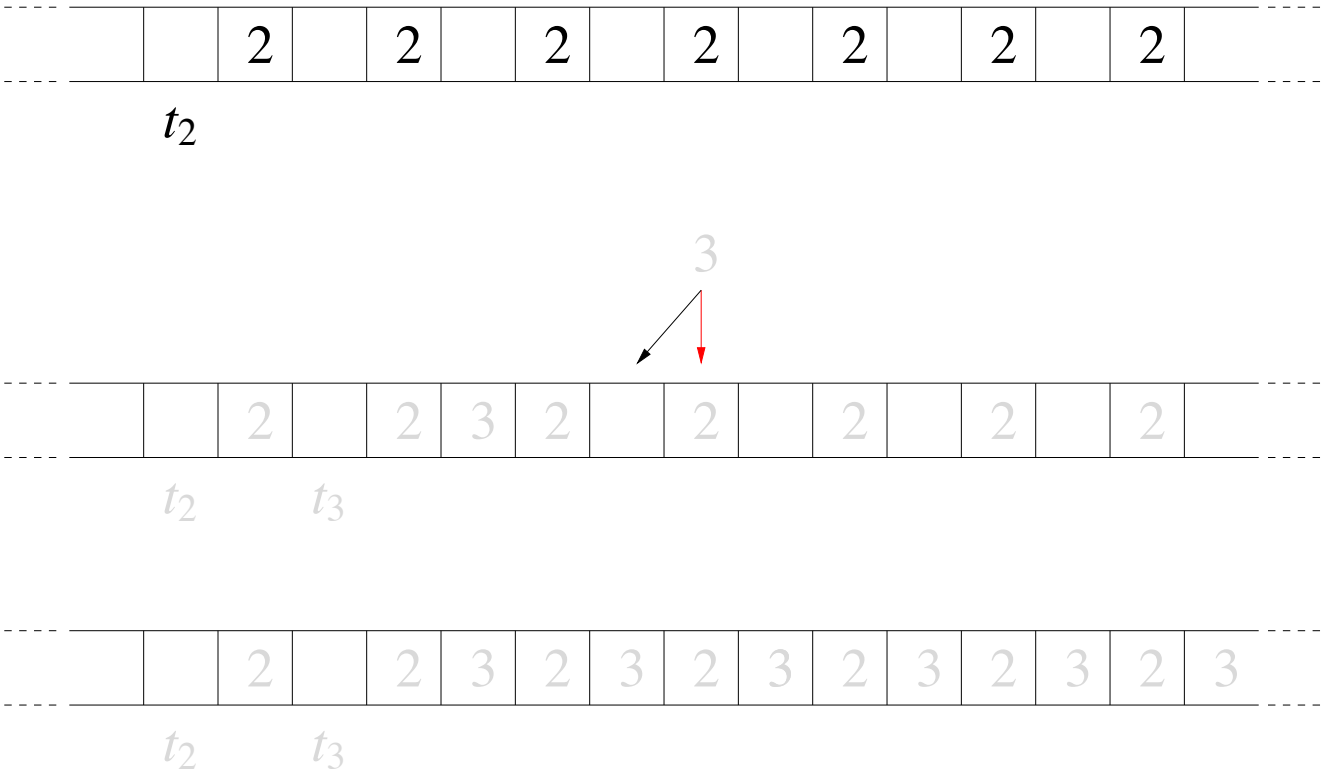
The Model

- Each client c_i characterized by
  - active cycle (arrival time t_i and departure time d_i)
  - laxity (window) w_i



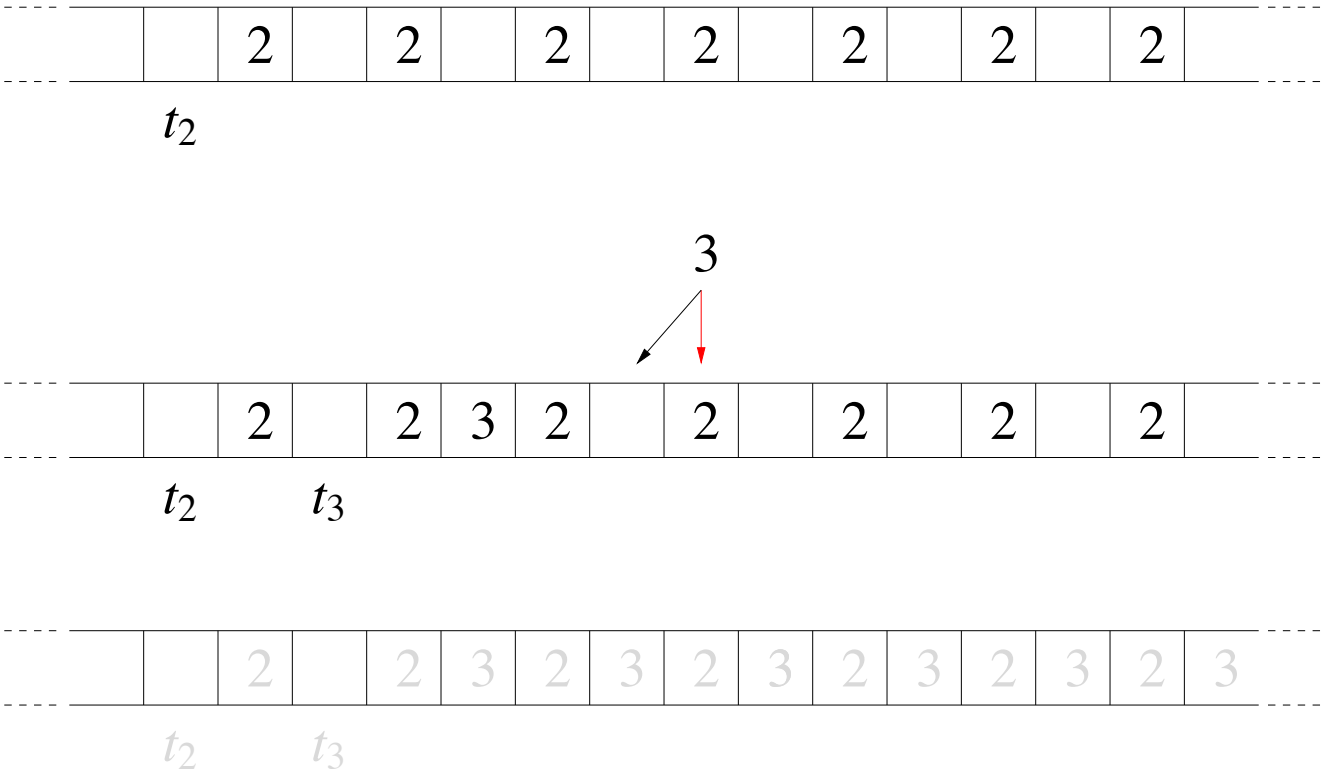
The Model

- A channel can receive only one transmission at a time!



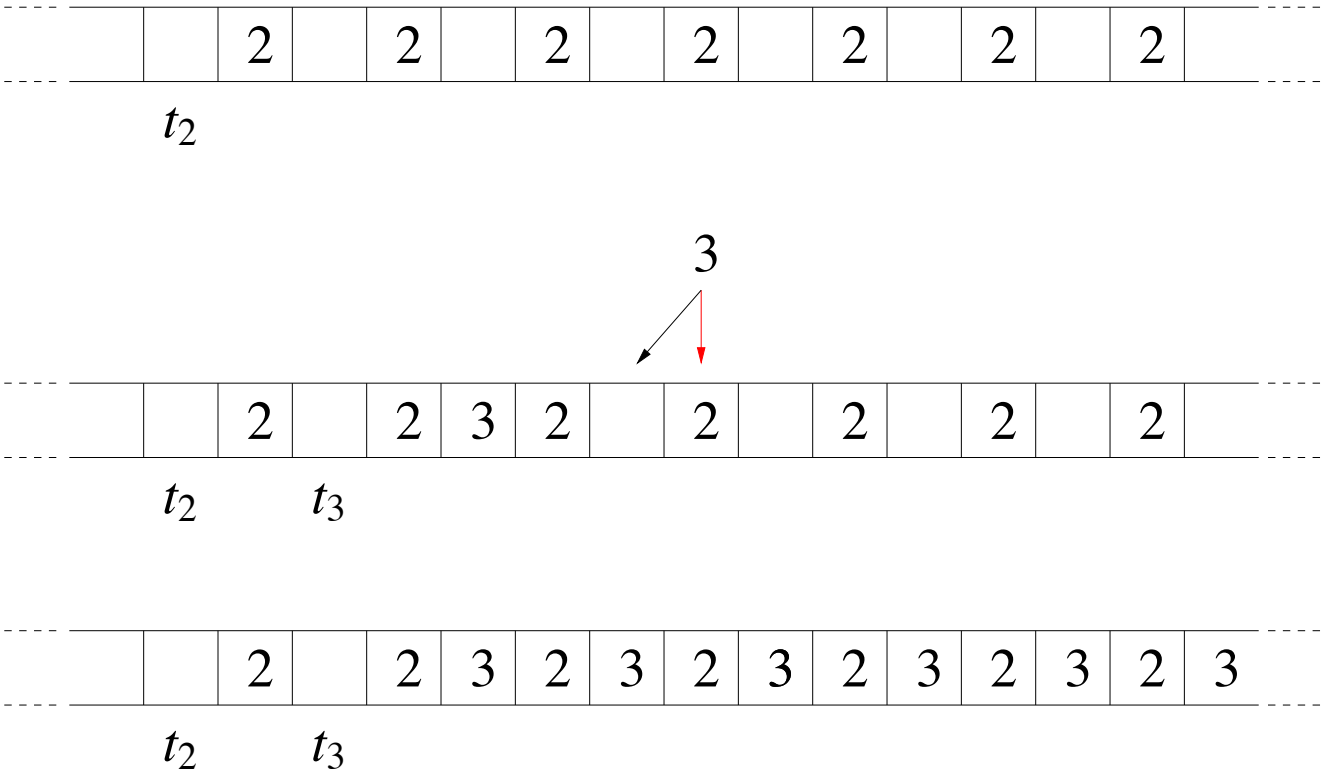
The Model

- A channel can receive only one transmission at a time!



The Model

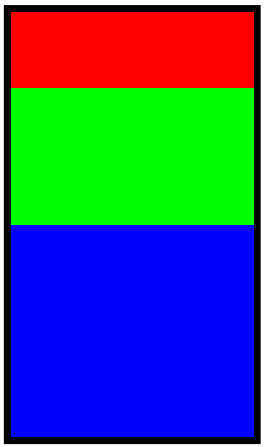
- A channel can receive only one transmission at a time!



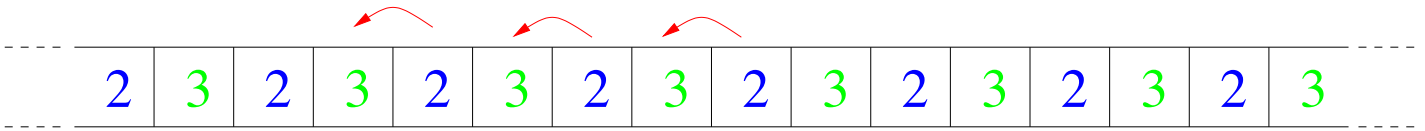
WS vs. UFBP

Why is WS more challenging?

UFBP:



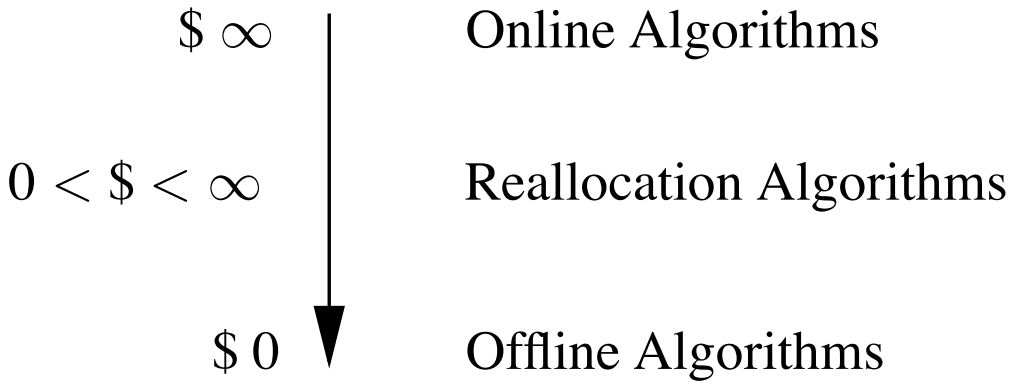
WS:



The Model

- Windows Scheduling (WS) (Bar-Noy et al. 03,07): clients do not leave.
- WS with Temporary Items (Chan-Wong 05): assignments are final.
- **WS with Reallocation (this work):**
  clients may be reallocated ... but at a cost!

Reallocation algorithms: middle ground between ...

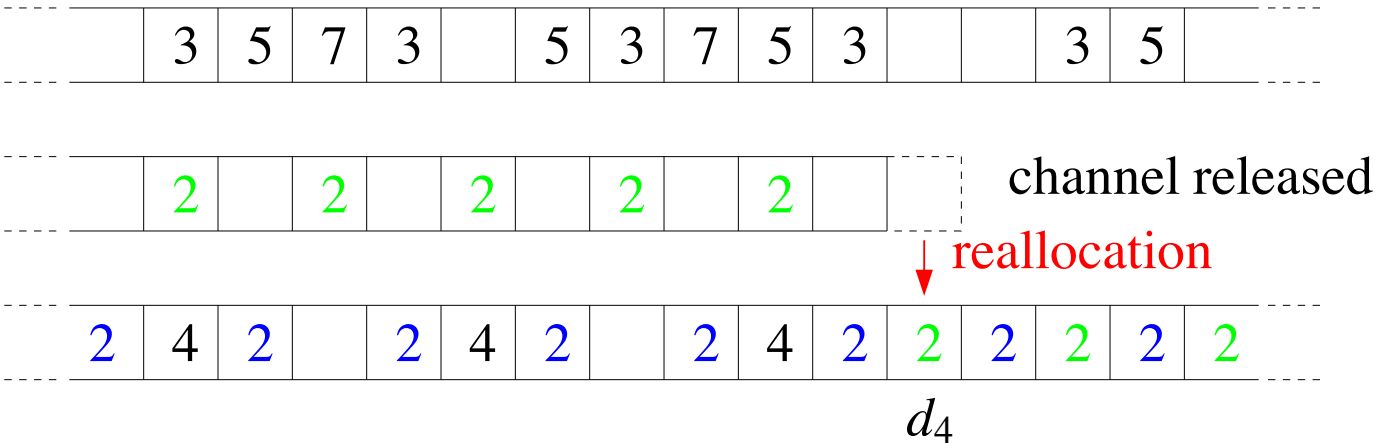


Reallocation also studied in Job Scheduling, Load Balancing, UFBP, etc.

The Assignment Problem

Given a set of clients and an infinite set of channels,
assign clients' transmissions to channels so that,

1. while client c_i is active $\exists \geq 1$ transmission from c_i to *some* channel scheduled in any w_i consecutive time slots.
2. there is at most one client assigned to each channel in each time slot.



Performance Metric

Competitive analysis:

- Bar-Noy et al. 03,07 (*against current load*):

$$\max_r \frac{ALG(r)}{OPT(r)}$$

- Chan-Wong 05 (*against peak load*):

$$\frac{\max_r ALG(r)}{\max_r OPT(r)}$$

- **this work (*against current load*):**

$$\max_t \left(\frac{R(r)}{r} + \frac{ALG(r)}{OPT(r)} \right)$$

Rounds defined by departures/arrivals.
$ALG(r) = \#$ channels used by ALG in round r.
$OPT(r) =$ minimum # channels needed in round r.
$R(r) = \#$ reallocations incurred by ALG up to round r.

Our Contribution

- Preemptive Reallocation:
  upon each arrival/departure: consolidate to good offline packing
- Lazy Reallocation:
  reallocate only when number of channels exceeds a threshold
- Classified Reallocation:
  classify clients by laxity (more later)
  first online WS protocol for
  dynamic scenarios (clients may leave)
  with theoretical guarantees (against current load)
- WS performance metric including reallocations
- simulations for all three protocols

Classified Reallocation

Main ideas:

- assume laxities powers of 2
- do not "mix" laxities,
  i.e., for any given channel, all clients assigned have the same laxity
- allow at most one non-full channel for each active laxity
  $\Rightarrow$ at most **log $\mathbf{w}_{\max}/\mathbf{w}_{\min}$ channels** sub-optimally used
  $\Rightarrow$ when a client leaves at most **one reallocation** is needed.
- what if $\log w_{\max}$ is large?
  $\Rightarrow$ have **one more channel** for all clients with $w \geq 2\lceil\lceil n\rceil\rceil$
  $\Rightarrow$ **linear reallocations only after n is doubled or halved**

Classified Reallocation

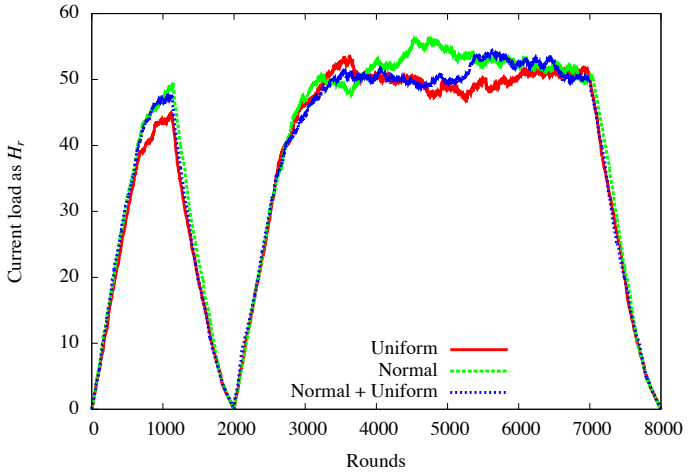
Theorem

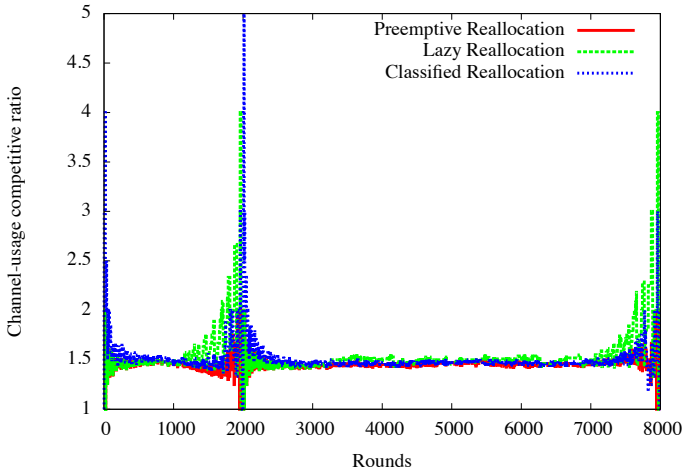
For any set of clients with laxity 2^i, $i = 0, 1, \ldots$, the schedule obtained by the Classified Reallocation algorithm requires at most $3r/2$ reallocations up to round r, and for any round r such that $n(r) > 0$ clients are active, the number of channels reserved is at most

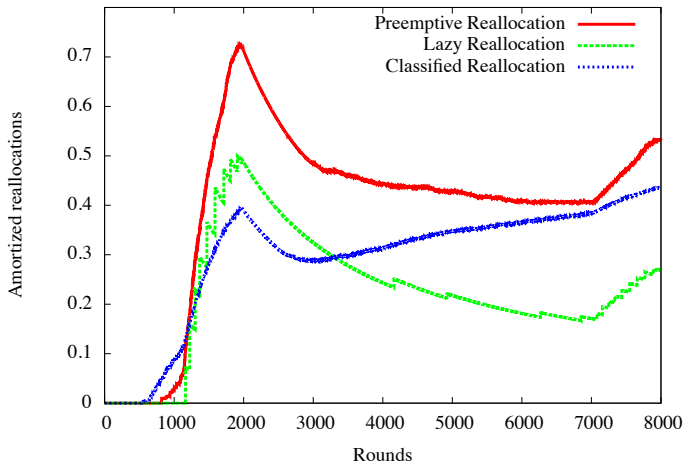
$$OPT(r) + 1 + \log\left(\frac{\min\{w_{\max}(r), \lceil\lceil n(r)\rceil\rceil\}}{w_{\min}(r)}\right)$$

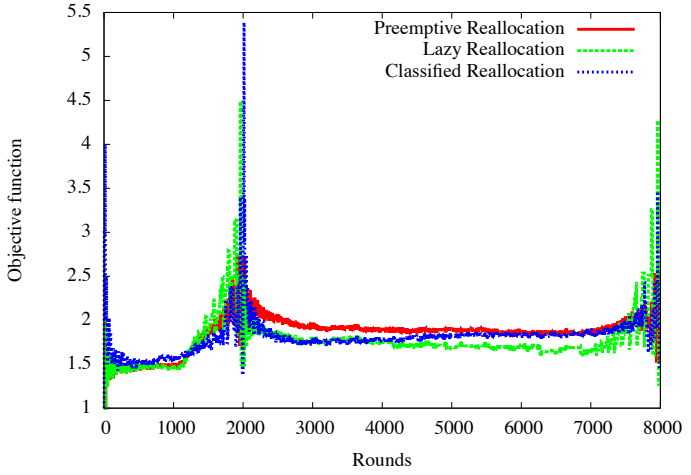
Simulations

Performance along rounds for normally distributed inputs.



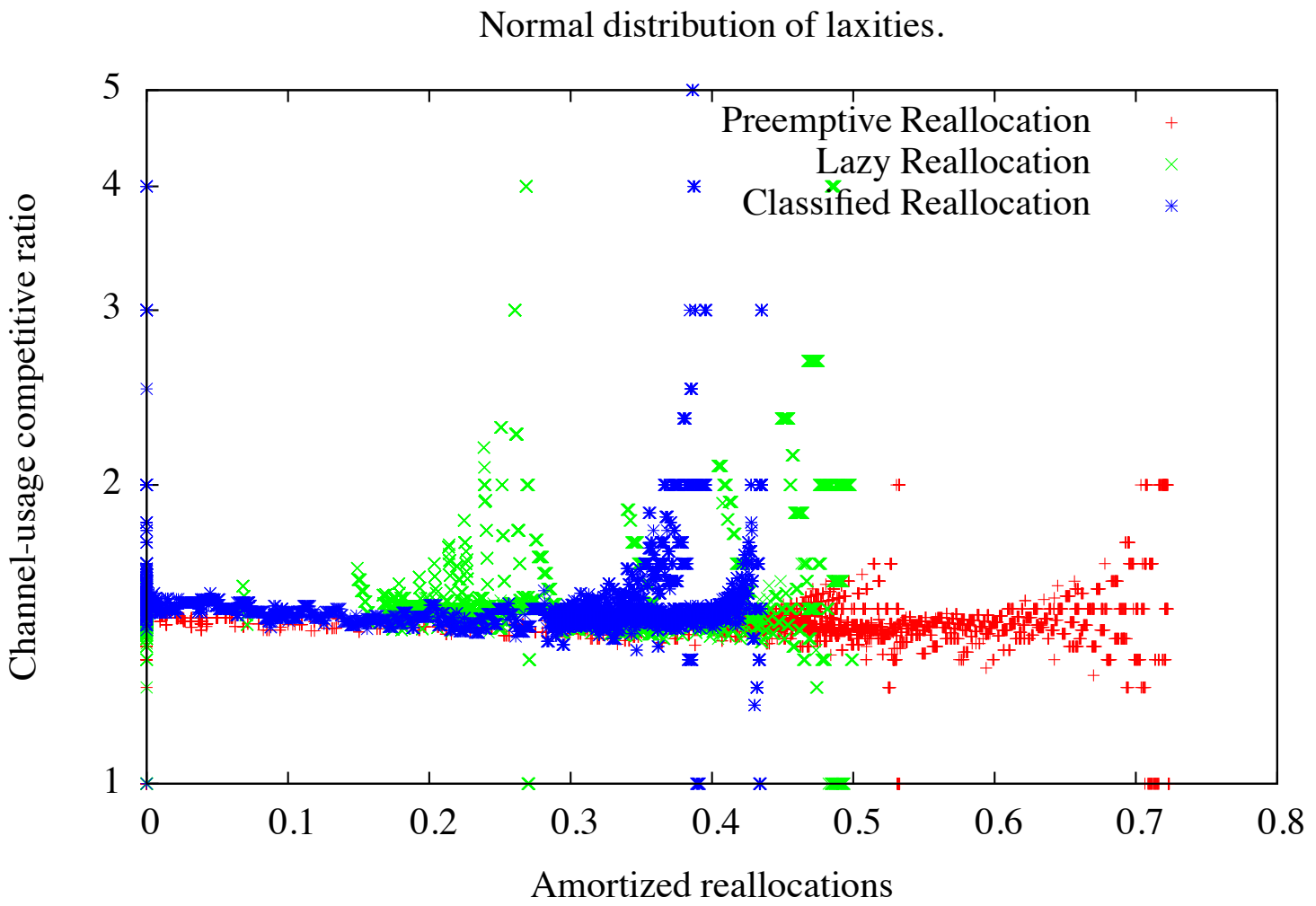






Simulations

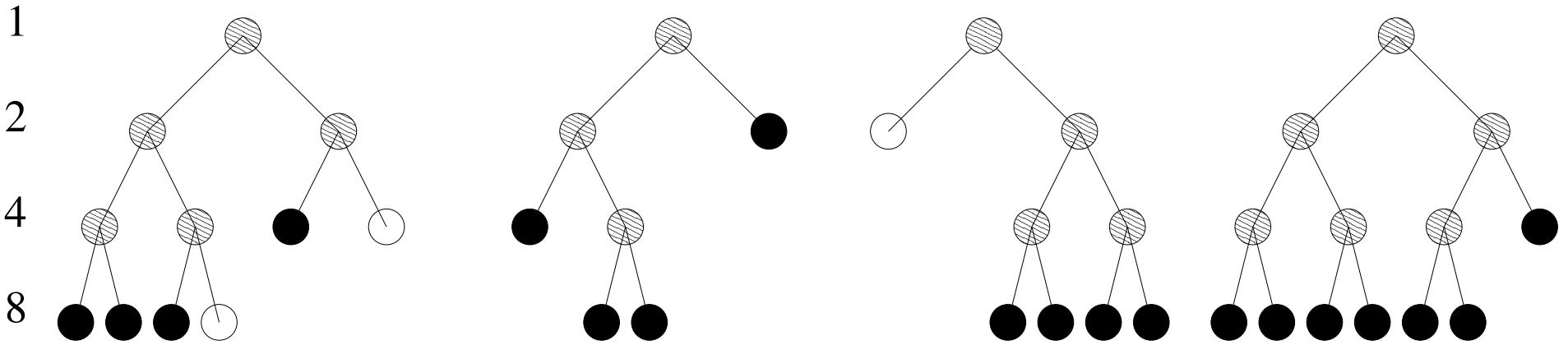
Channel usage vs. reallocations.



Thank you

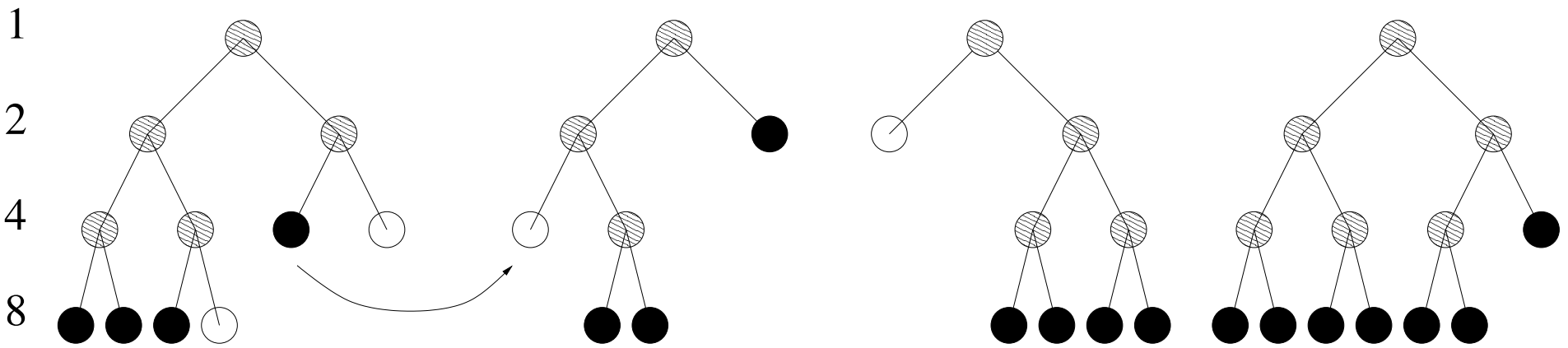
Preemptive Reallocation

Use *packed broadcast trees*:



Preemptive Reallocation

Use *packed broadcast trees*:



Preemptive Reallocation

Use *packed broadcast trees*: