Distributed Station Assignment through Learning

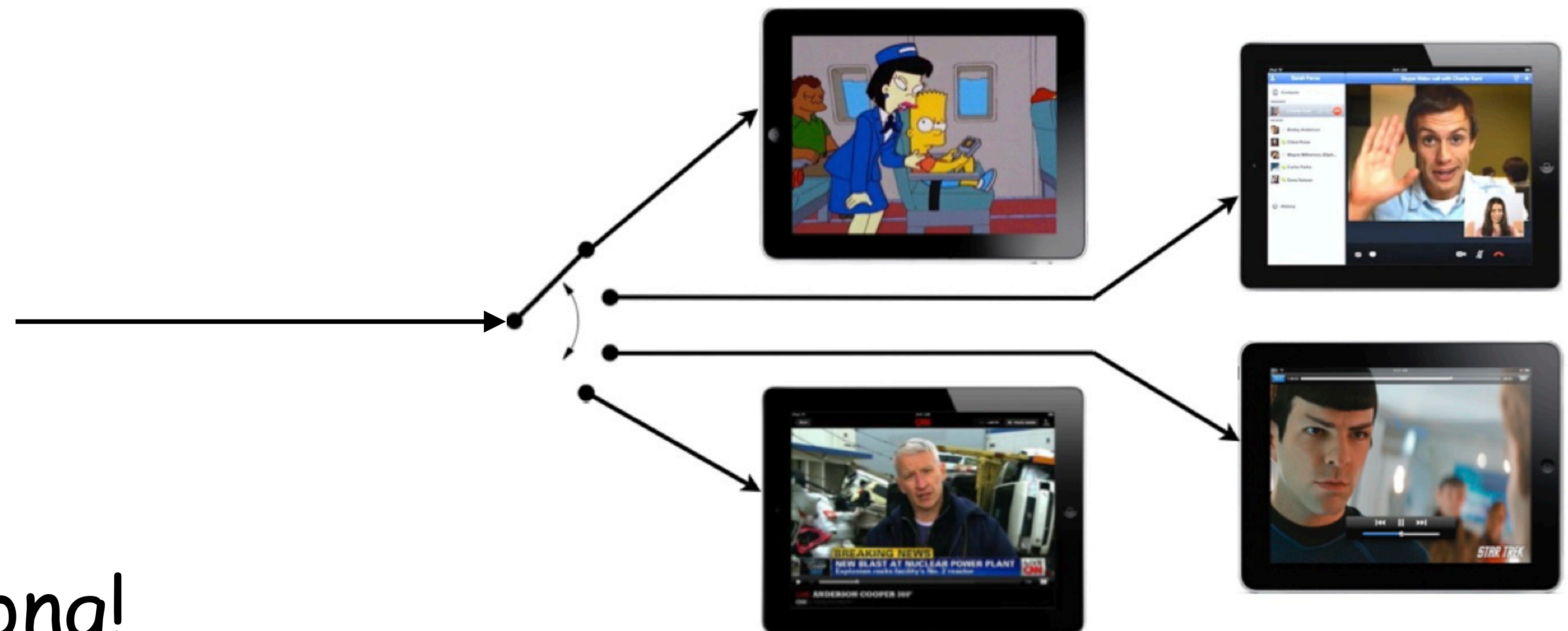
Lu Dong   Miguel A. Mosteiro   Michelle Wang

Dept. of Computer Science, Pace University, New York, NY, USA

NETYS 2024

Station Assignment Motivation

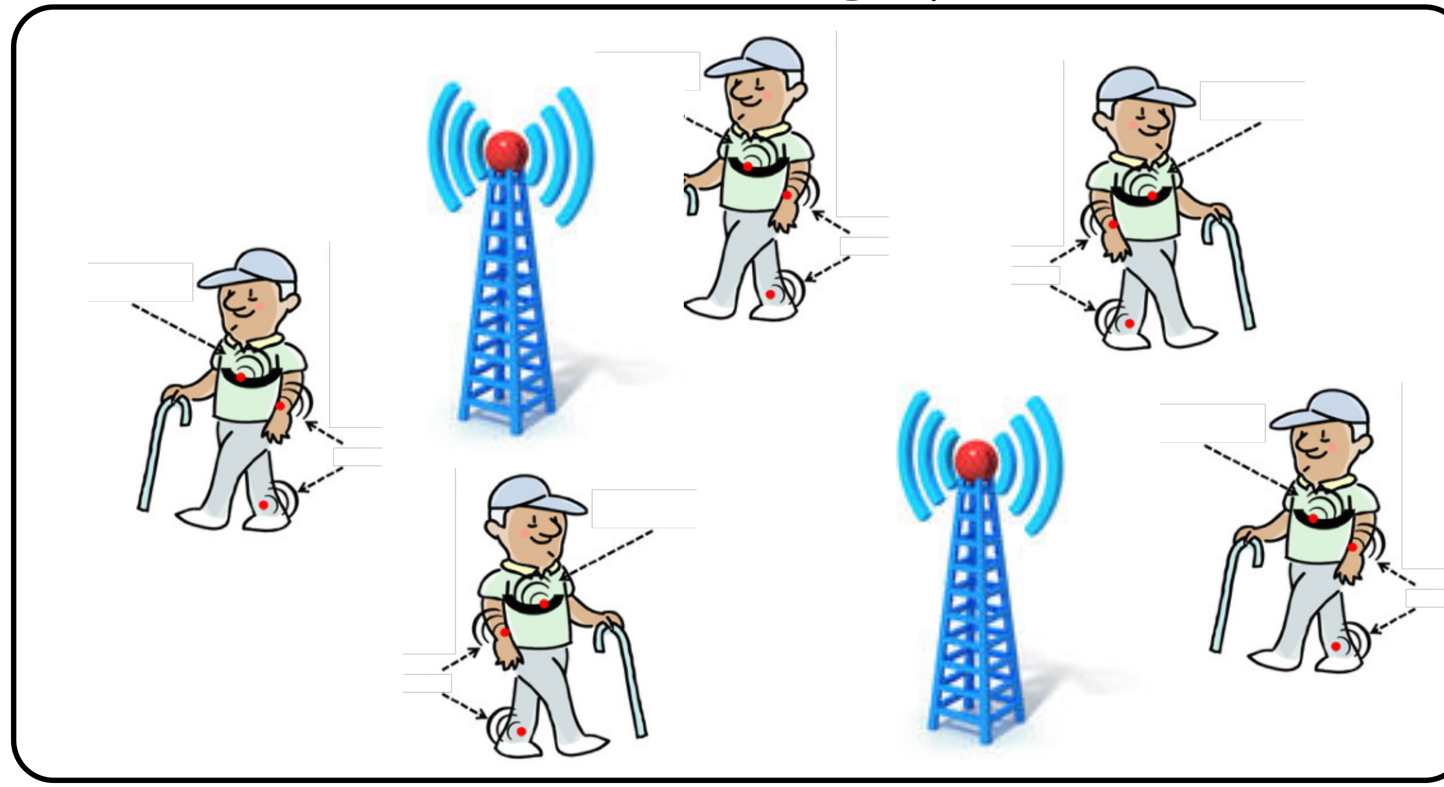
Multiple users need access to a shared resource
each user can wait for a while ...



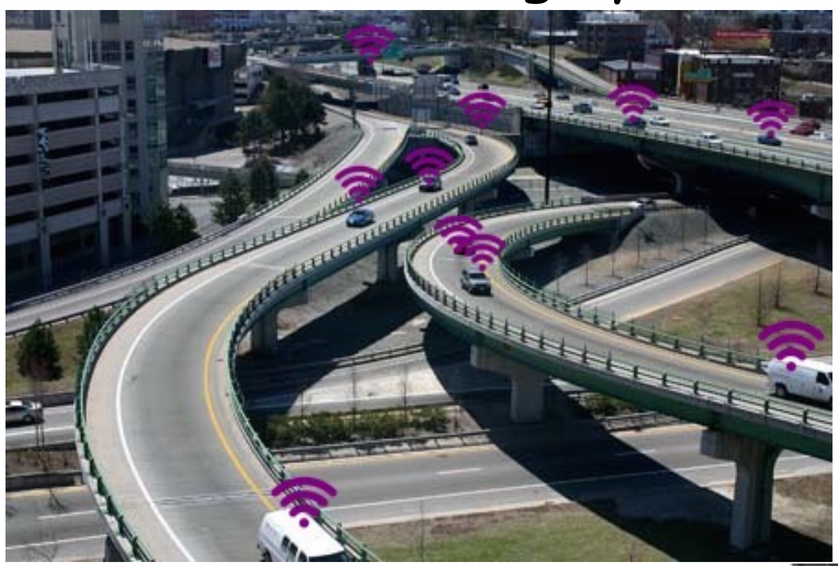
but not too long!

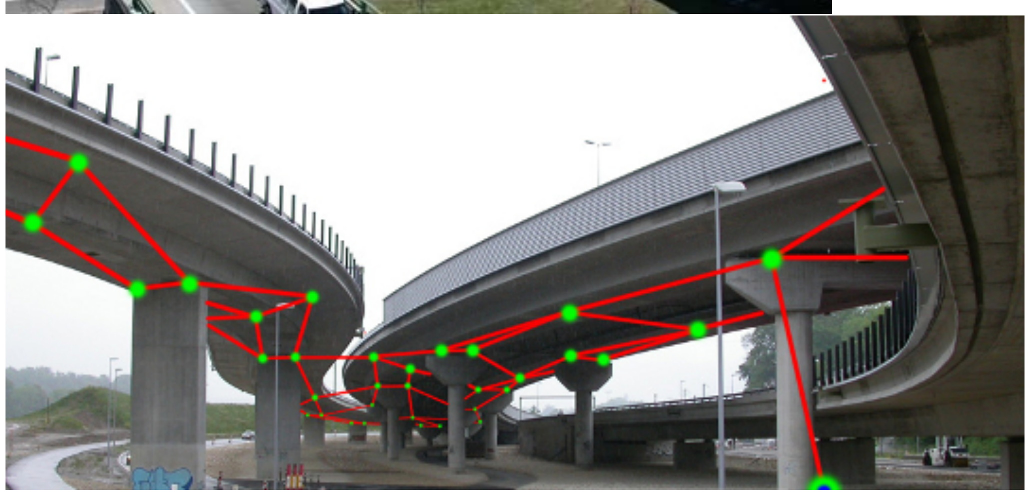
Station Assignment Applications

Wearable health-monitoring systems



Traffic monitoring systems





Inventory replenishment

Dynamic Allocation Problem

<u>Radio Network</u>:

- A set of static stations
- A set of mobile clients

To upload (or download) packets,

Clients are allocated to Stations,

but there are restrictions...

Model

Slotted time.

Client c:

- laxity w_c:

  c must transmit to some station at least one packet within each w_c consecutive time slots while active.

- bandwidth requirement b_c

Station s:

- bandwidth capacity B_s:

  maximum aggregated bandwidth of clients that may transmit to s in each time slot.

Station Assignment Problem (SA)

Given a set of clients and set of stations,
assign clients' transmissions to stations so that:

1) Each client c transmits to a station at least once within each w_c time slots.

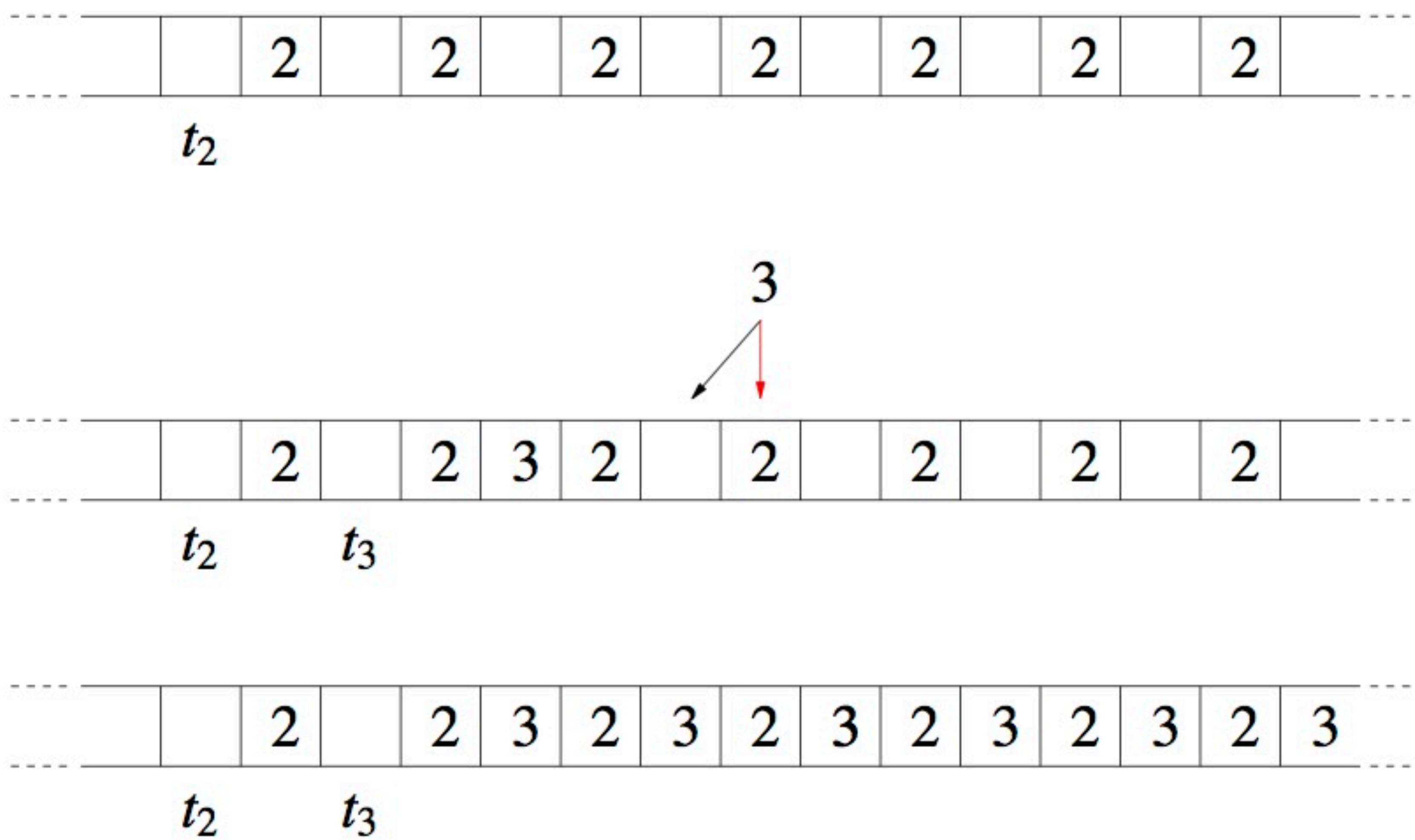
2) In each time slot, each s receives from a set of clients whose aggregated b_c is at most B.

… minimizing resouces utilized.

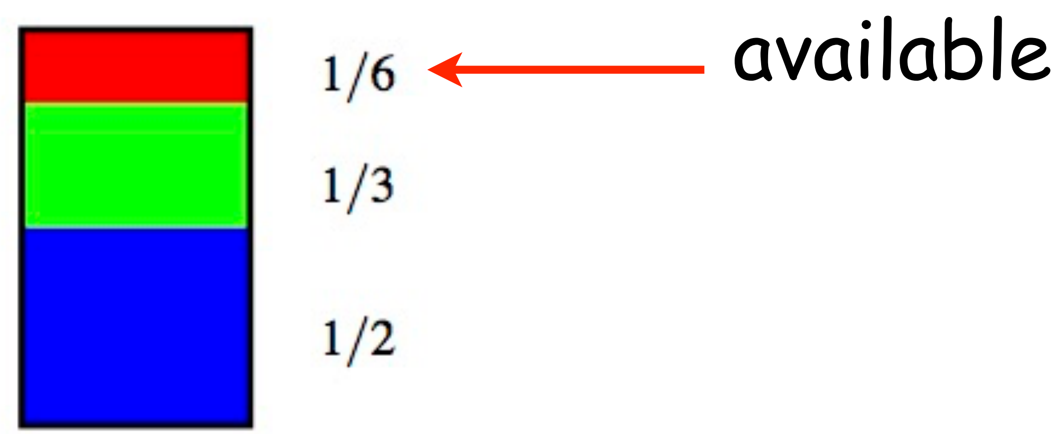
SA Problem

$b_2 = B$, $w_2 = 2$

$b_3 = B$, $w_3 = 3$



SA is more restrictive than UFBP...



Models

Centralized, $b_c=B$:

Windows Scheduling (WS) [Bar-Noy et al.,03 & 07]: clients do not leave.

WS with Temporary Items [Chan,Wong,05]: allocations are final.

WS [Farach-Colton et al.,14]: with reallocation at constant cost (1).

Centralized, $b_c \leq B$:

SA [Fernandez-Anta et al.,13]: no reallocation.

SA [Halper et al.,15]: with reallocation at proportional cost (ρ/w_c), reallocation + channel-usage performance metrics (≡ 1 station, unbounded channels).

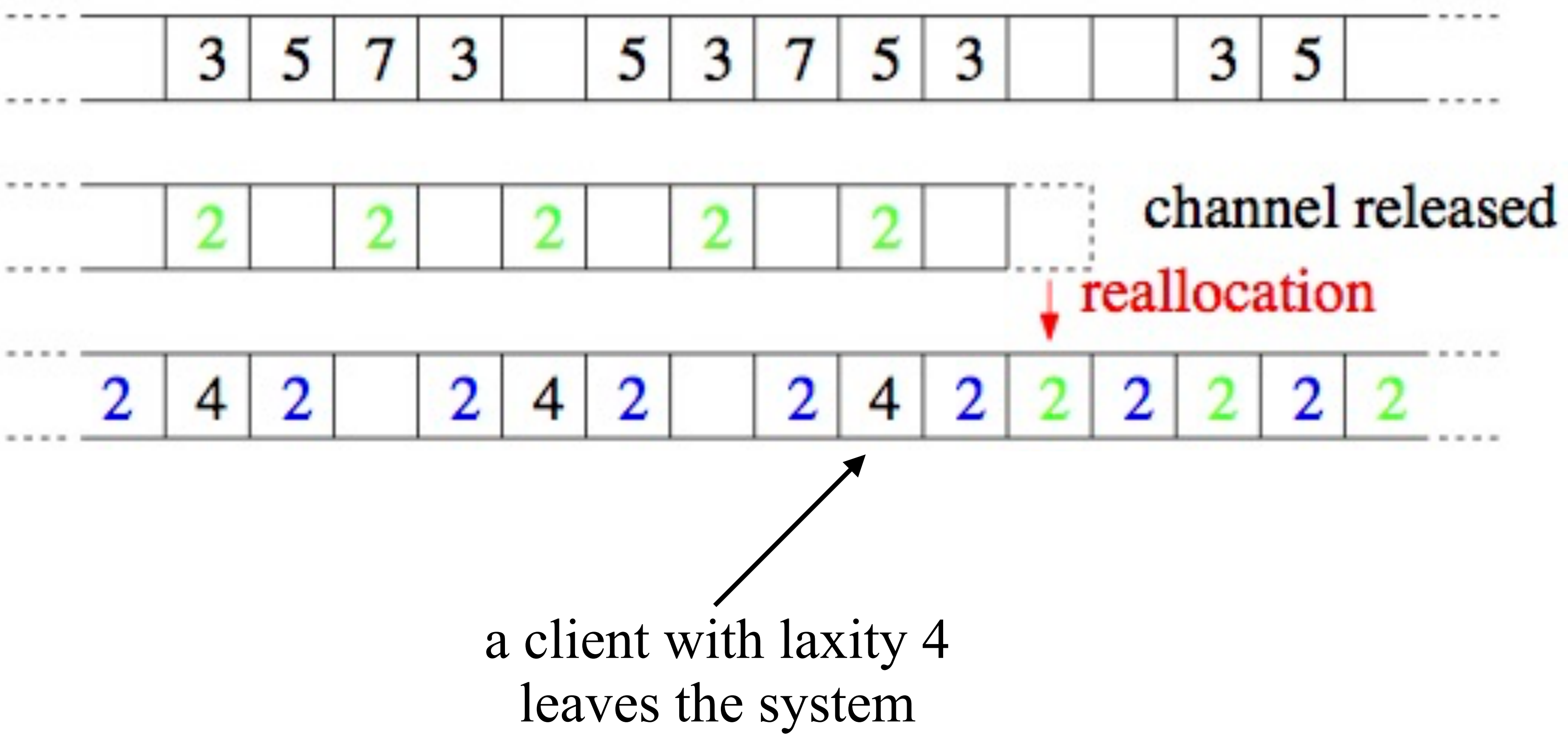
<u>This paper</u>: Distributed $b_c \leq B$:

SA through Learning: with reallocation at proportional cost (ρ/w_c), reallocation + channel-usage + energy performance metrics (set of stations, unbounded channels).

Reallocation Algorithms

Middle-ground between
online algorithms (infinite cost reallocations)
and offline algorithms (free reallocations).

Example: $b_c=B$



WS-SA Reallocation Algorithms

[Farach-Colton et al., 14]

- Centralized Preemptive Reallocation: low channel usage.
- Centralized Classified Reallocation: low reallocation cost.

[Halper et al., 14]

- Centralized Classified Preemptive Reallocation:

  trade-offs between low channel usage and low reallocation cost.

This paper:

New approach: Distributed Learning Reallocation Algorithms

Multi-Agent Reinforcement Learning (MARL) with

Independent Proximal Policy Optimization (IPPO)

Performance Metrics

- [Halper et al., 15]:

$$\max_{r:E(ALG,r)\neq\emptyset} \frac{\mathscr{H}(ALG,r)}{\mathscr{H}(OPT,r)} \leq \alpha$$

$$\max_{r:R(ALG,r)\neq\emptyset} \frac{\mathscr{R}(ALG,r)}{\mathscr{D}(ALG,r)} \leq \beta$$

- <u>This paper</u>: additionally

$$\max_{r:E(ALG,r)\neq\emptyset} \frac{\mathscr{E}(ALG,r)}{\mathscr{E}(OPT,r)} \leq \gamma$$

(α,β,γ)-approximation against current load

$\mathscr{H}$: number of channels used.

$\mathscr{R}$: cost of reallocations.

$\mathscr{D}$: weight of departed clients.

$\mathscr{E}$: energy consumed by clients.

$$\mathscr{H}(OPT,r) \geq \left\lceil \sum_c B/(b_c w_c(r)) \right\rceil$$

$$\mathscr{E}(OPT,r) \geq \sum_c \epsilon \min_s d(c,s,r)^\delta / w_c(r)$$

$$\mathscr{R}(ALG,r) = \sum_{c\in R(ALG,r)} \rho / w_c(r)$$

$$\mathscr{D}(ALG,r) = \sum_{c\in D(ALG,r)} 1/w_c$$

Distributed Learning Reallocation

- In each control sub-round:
  - each client
    - exchanges information to decide whether to upload this round and to which station,
    - broadcasts ID of chosen station,
  - each station
    - activates/deactivates channels and reallocates among channels according to ID's received.
- In each data sub-round:
  - each client uploading transmits a packet to chosen station.

MARL Formal Framework

Decentralized Partially Observable Markov Decision Process (Dec-POMDP)

$$\langle V_c, S, A, P_S, O, P_O, R, W \rangle$$

reward function R(c,r) for client c after action taken in round r

if c uploads:

$$\Gamma - \left[\frac{|s(c,r) - s(c,r-1)|}{|V_s|} \right] \frac{\rho}{w_c(r)} - \eta \cdot X(s(c,r),r) - \epsilon \cdot d(c,s,r)^{\delta}$$

reallocations cost

channel usage cost

energy cost

if c does not upload:

$$\Gamma - \xi \cdot w_c(r) \qquad \text{if } w_c(r) > w_c/\kappa$$

$$\Gamma - \xi / w_c(r) \qquad \text{otherwise}$$

we need to learn a policy to maximize this reward

Policy Optimization

Goal: learn a policy to maximize expected reward.

Our state space is too large (locations),

⇒ compute exact action-value function (Q) and/or state-value function (V) is time consuming,

⇒ we use instead a policy gradient method to estimate an advantage-value function A=Q-V.

Policy Optimization

Independent Proximal Policy Optimization (IPPO):

[Schulman et al.,17 & de Witt et al.,20]

$\Rightarrow$improve stability avoiding change policy too much:

$$\pi_\theta = \arg\max_{\pi_\theta} \hat{\mathbf{E}}_r \left[L(\pi_\theta, \pi_{\theta_{old}}, a_r(c), s_r(c)) \right]$$

policy

by stochastic gradient ascent

empirical average

advantage function estimated as in [Schulman et al.,15]

$$\begin{cases} \min\left(\dfrac{\pi_\theta(a_r(c) \mid s_r(c))}{\pi_{\theta_{old}}(a_r(c) \mid s_r(c))}, 1+\epsilon\right)\hat{A}_r(c), & \text{if } \hat{A}_r(c) \geq 0 \\[2ex] \max\left(\dfrac{\pi_\theta(a_r(c) \mid s_r(c))}{\pi_{\theta_{old}}(a_r(c) \mid s_r(c))}, 1-\epsilon\right)\hat{A}_r(c), & \text{otherwise} \end{cases}$$

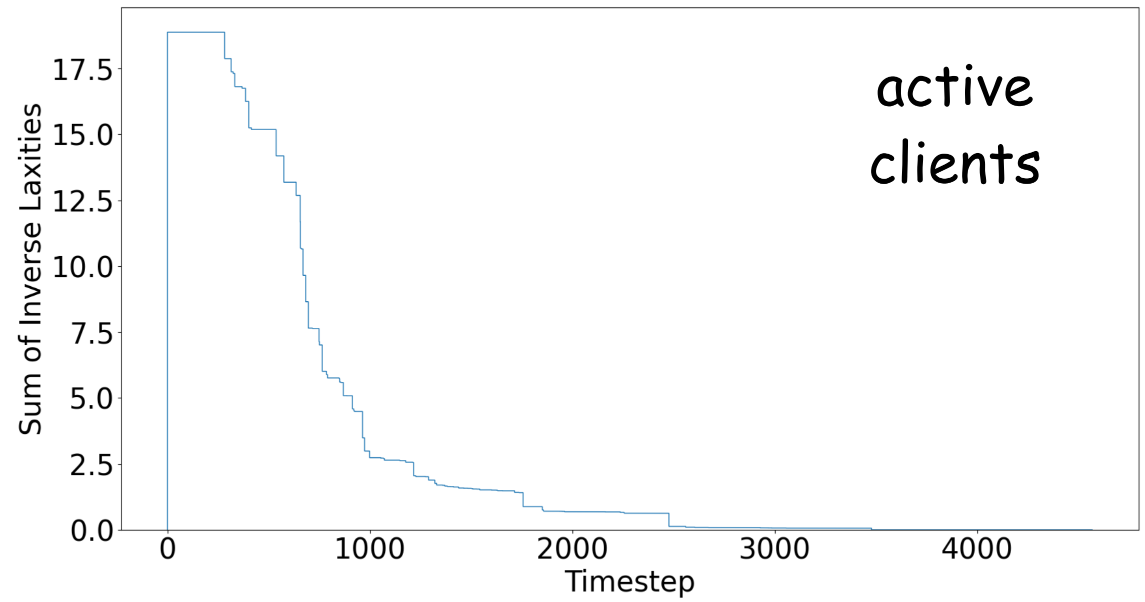
SA Protocol

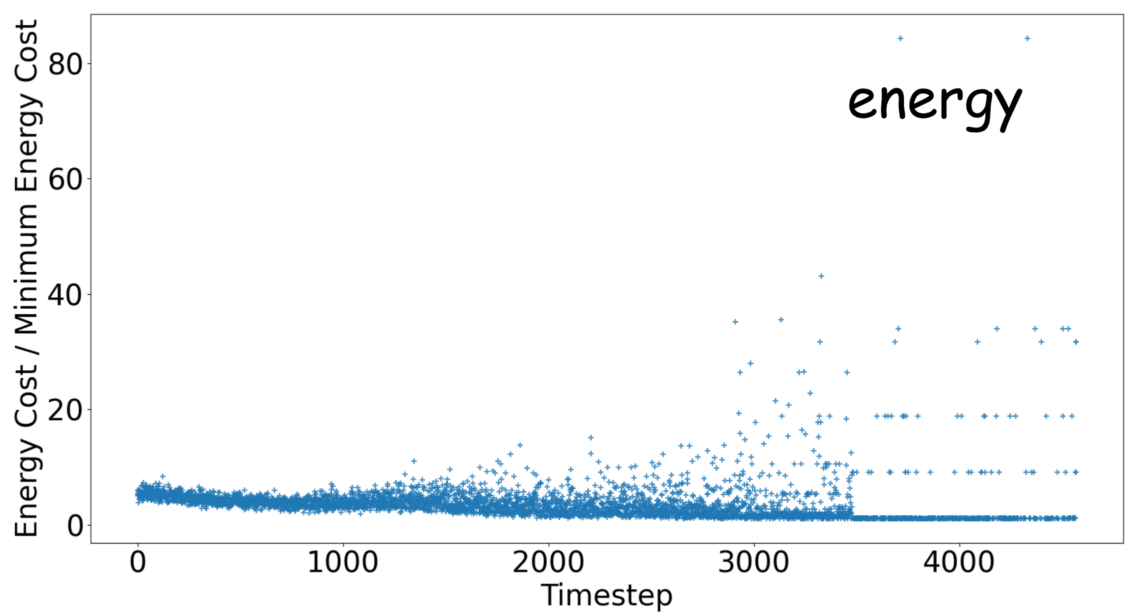
Algorithm 1: SA protocol for each client $c \in V_c$. $Coord_\sigma$ are the location coordinates of station σ. $X(\sigma)$ is the value of the indicator variable $X(\sigma(c,r),r)$. w_c, b_c are as defined in the model section. T is the parametric number of iterations between policy updates (a.k.a. minibatch size).

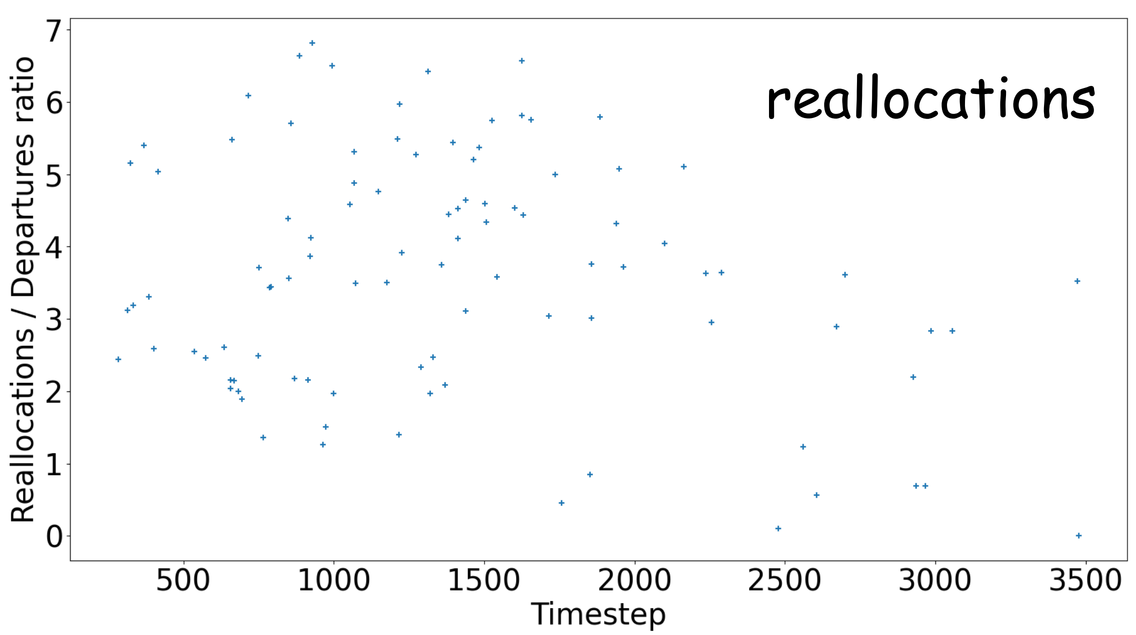
$\sigma_{prev} \leftarrow 0$
$w_{left} \leftarrow w_c$
$\pi \leftarrow$ uniform distribution over integers in $[0, m]$
$i \leftarrow 1$ `// Minibatch iteration counter`
for $r = 1, 2, \ldots$ **do**
    `// control subround`
    $x \leftarrow$ choose a number in $[0, m]$ at random with probability distribution π
    if $x \neq 0$ **then**
        broadcast $\langle c, w_c, b_c, x \rangle$
        receive $\langle \sigma, Coord_\sigma, X(c, \sigma) \rangle$ from station $\sigma = x$
    $R_i \leftarrow$ compute reward using $Coord_\sigma, X(\sigma), \sigma_{prev}, w_{left}$ and x
    `// Equations 1 and 2`
    if $i = T$ **then**
        compute advantage estimators $\hat{A}_1, \ldots, \hat{A}_T$ using $R_1, \ldots, R_T$
        `// Equation 4 in [20]`
        update π `// Equation 3`
        $i \leftarrow 0$
    $i \leftarrow i + 1$
    `// data subround`
    if $x \neq 0$ **then**
        upload to station x
        $\sigma_{prev} \leftarrow x$
        $w_{left} \leftarrow w_c$
    else
        $w_{left} \leftarrow w_{left} - 1$

Simulations

$|V_c|=100$, $|V_s|=10$, $w_c=2^{(\text{random})}$, $b_c=B$, $\epsilon=1$, $\rho=1$, $\eta=1$, $\xi=1$, $\delta=2$







With respect to previous centralized scheduler, similar reallocations ratio with a distributed scheduler. First energy evaluation.

Thank you!

Miguel A. Mosteiro
(mmosteiro@pace.edu)