Stable Resource Allocation Policies for Fading Channels

Atilla Eryılmaz, R. Srikant and J.R. Perkins

Outline

- System Model & Objective
- A set of Resource Allocation Policies
- Stability of the Stochastic model
- Applications
  - $\longrightarrow$ Imperfect Queue length information
  - $\longrightarrow$ Reducing computational complexity
- Conclusions

System Model

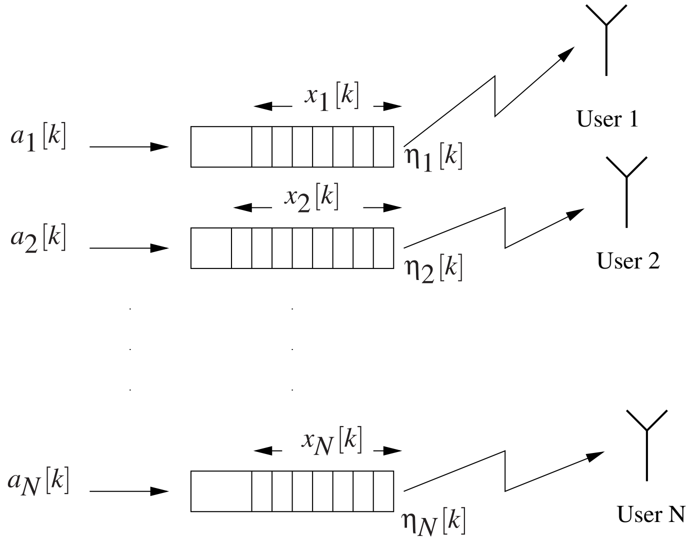


Figure 1: Downlink model

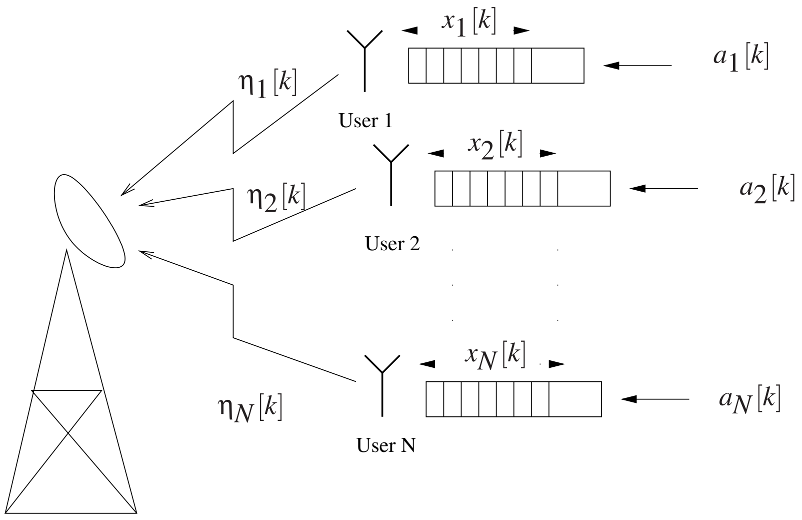
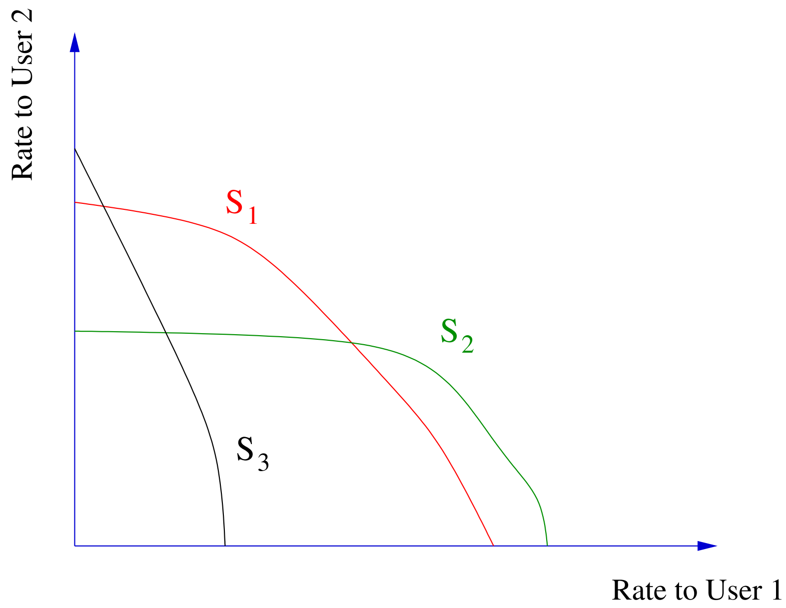


Figure 2: Uplink model

- N queues of infinite length, each maintained for a different user. Time is slotted.
- $x_i[k]$: length of queue i at the beginning of slot k.
- $a_i[k]$: number of bits that arrive at queue i at slot k.
- $\eta_i[k]$: number of bits of service offered to queue i in slot k.
- $x_i[k+1] = (x_i[k] + a_i[k] - \eta_i[k])^+$, where $(y)^+ = y$ if $y > 0$ and zero otherwise.

Assumptions - Channel state process

- Channel is in one of J states
- It has a stationary distribution $\{\pi_j\}$, that is unknown.



$\rightarrow$ $s[n]$ = the state of the channel at time n.

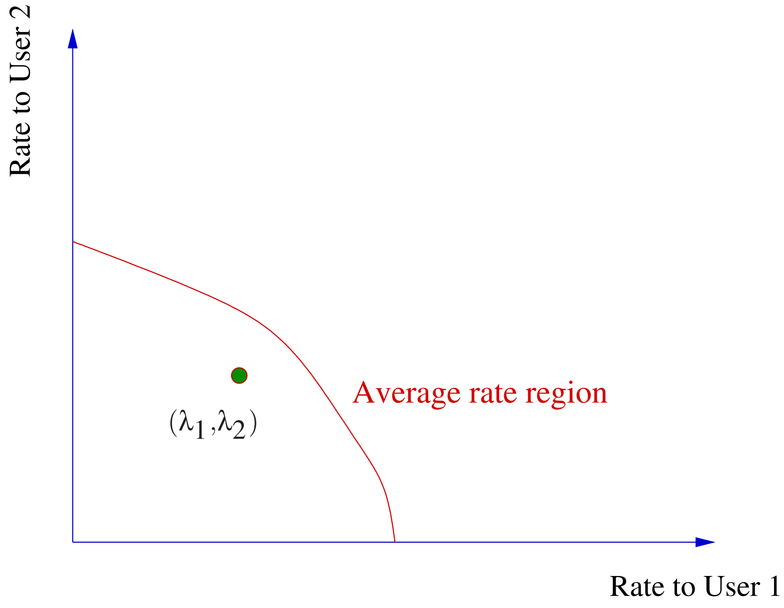
$\rightarrow$

$$\lim_{M\to\infty} E\left[\left|\pi_j - \frac{1}{M}\sum_{n=k}^{k+M-1} I_{s[n]=j}\right|\right] = 0,$$

for any $k > 0$ and $j = 1, 2, \ldots, J$.

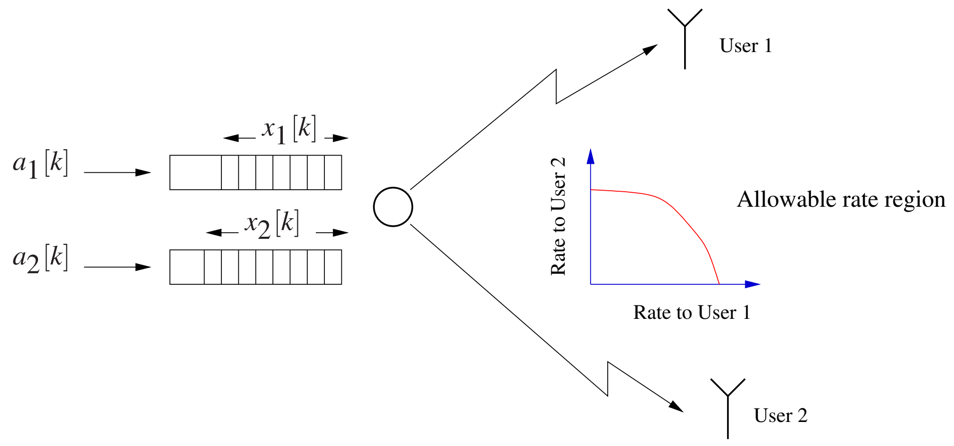
Assumptions - Arrival process

- For each queue arrivals form a stationary process with mean $\lambda_i = E[a_i[1]]$.
- The vector of mean arrival rates $\vec{\lambda} := (\lambda_1, \ldots, \lambda_N)'$ lies inside average allowable rate region, $\bar{\mathcal{S}}$.



$$\lim_{M\to\infty} E\left[\left|\frac{1}{M}\sum_{n=k}^{k+M-1} a_i[n] - \lambda_i\right|\right] = 0, \qquad \forall i.$$

Goal



- Our goal is to find a

  RESOURCE ALLOCATION STRATEGY THAT STABILIZES THE SYSTEM

  using only queue length information and the current channel state (i.e., *without knowing channel or arrival statistics.*)

- We also aim to incorporate imperfect queue length information and other approximations that can simplify the implementation of our policy.

Resource Allocation Policy

At any time k, given the current channel state $s[k]$ and the queue length vector $\vec{x}[k]$, the controller chooses a service rate vector $\vec{\mu} = (\mu_1, \cdots, \mu_N)' \in \mathcal{S}_{s[k]}$ that satisfies:

$$\vec{\mu} \in \arg\max_{\vec{\eta} \in \mathcal{S}_{s[k]}} \sum_{i=1}^{N} f_i(x_i[k])\eta_i, \tag{1}$$

where $f_i : [0, \infty) \longrightarrow [0, \infty)$ are nondecreasing, continuous functions with $\lim_{x \to \infty} f_i(x) = \infty$.
$\longrightarrow$ Some examples of the the functions $f_i(\cdot)$ are $f_i(x) = (K_i x)^\alpha$ for any $K_i \in [0, \infty)$ and $\alpha \in [0, \infty)$, or $f_i(x) = e^{\sqrt{x}}$.

$$\max_{\vec{\eta} \in S_{s[k]}} x_1[k]\eta_1 + x_2[k]\eta_2$$

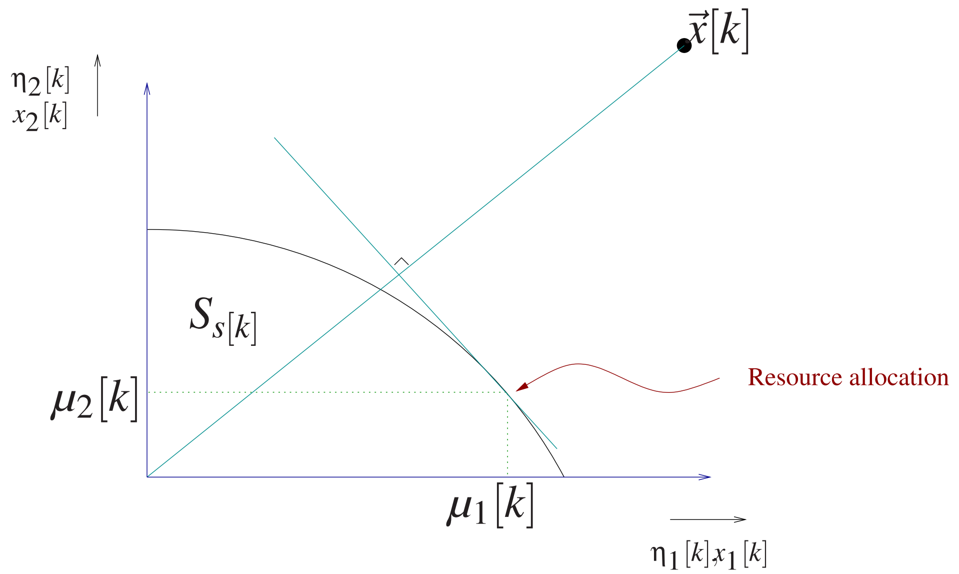


Figure 3: An implementation of the policy for $f_i(x) = x$

- Policy defined by (1) may be difficult to implement due to various reasons such as
  - $\longrightarrow$ queue length information may be delayed - Uplink model
  - $\longrightarrow$ maximization may be too complex - if optimization is over a large set.
  - $\longrightarrow$ waiting times instead of the queue lengths $\vec{x}[k]$ may be used in the equation

Resource Allocation Policy [Cont'd]

We consider a relaxed version of the policy, satisfying the following property:

Property 1 *Given any* ρ *and* ζ *such that* $0 \leq \rho, \zeta < 1$, *there exists a* $B > 0$ *such that the resource allocation policy satisfies the following condition: at any time* k, *with probability greater than* $(1-\rho)$, *the controller chooses a service rate vector* $\vec{\mu} = (\mu_1, \cdots, \mu_N)' \in \mathcal{S}_{s[k]}$ *that satisfies:*

$$\sum_{i=1}^{N} f_i(x_i[k])\mu_i \geq (1-\zeta) \max_{\vec{\eta} \in \mathcal{S}_{s[k]}} \sum_{i=1}^{N} f_i(x_i[k])\eta_i \tag{2}$$

whenever $\|\vec{x}[k]\| > B$. ◇

Stability of the stochastic model

- Main result is provided in the following theorem

Theorem 1 *For sufficiently small values of* $\zeta, \rho \geq 0$, *the system is stable in the mean under the policy described in previous slides, i.e.,*

$$\limsup_{p\to\infty} \frac{1}{p} \sum_{k=0}^{p-1} E\left[||\vec{f}(\vec{x}[k])||_2 \right] < \infty, \tag{3}$$

where $||\vec{f}(\vec{x})||_2 := \left(\sum_{i=1}^{N} f_i^2(x_i) \right)^{1/2}.$

Infrequent or delayed queue length updates

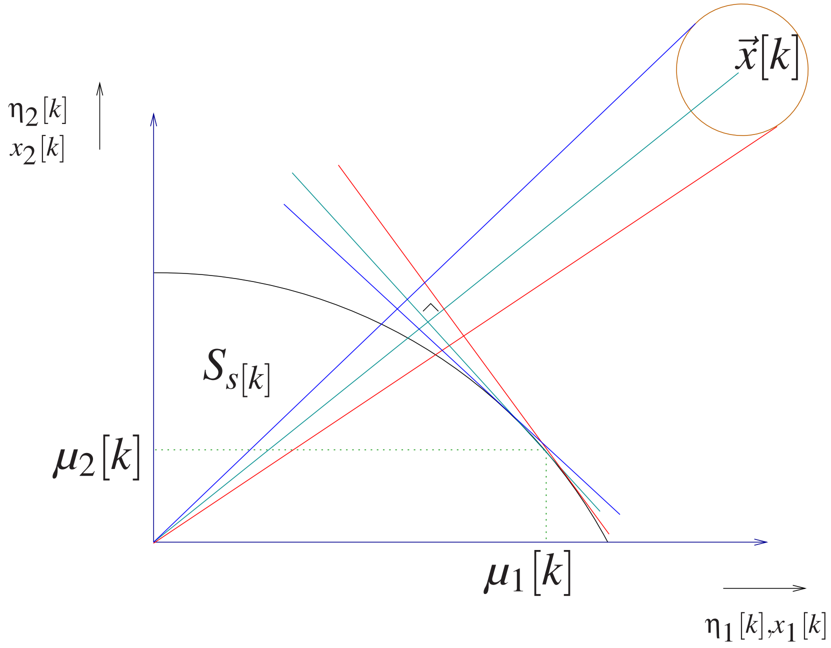
- Useful in the uplink scenario, where each user updates its queue length to the base station, once every T slots.
- Let $\hat{x}_i[k]$ denote the estimate of the queue length of the i^{th} queue at time k. In other words, $\hat{x}_i[k]$ is the last update of the queue length, prior to time k, received by the base station from Transmitter i.
- At each time slot k, the base station allocates a service rate vector that satisfies

$$\arg \max_{\vec{\eta}[k] \in S_{s[k]}} \sum_{i=1}^{N} f_i(\hat{x}_i[k]) \eta_i[k]. \tag{4}$$

Infrequent or delayed queue length updates[Cont'd]

Theorem 2 *Suppose that the base station is only allowed to sample the queue length information once every T slots and it uses this sampled value as the current queue length to determine the service rates according to (4), then the system is stable-in-the-mean.*

→ True, because with high probability, for large enough queue lengths, the estimate and actual values will be within a bounded distance from each other.



Reducing computational complexity

- Assuming that the set of allowable rates at each state has a finite size, consider the following algorithm (a modified version of Tassiulas', Shah and Prabhakar's algorithms for switches):

Algorithm A: Assume that the channel is time invariant. Let $\vec{\eta}[k]$ denote the rate vector used at time k. Then the algorithm is comprised of the repetition of the following steps:
(1) $\vec{h}[k] =$ next rate vector visited by the Hamiltonian walk.
(2)

$$\vec{\eta}[k] \in \arg \max_{\vec{\nu} \in \{\vec{\eta}[k-1], \vec{h}[k]\}} \sum_{i=1}^{N} f_i(x_i[k]) \nu_i .$$

$\longrightarrow$ Note that the optimization is only over two rate vectors.

Reducing computational complexity [Cont'd]

Theorem 3 *The policy defined by* Algorithm A *satisfies Property 1 defined earlier and hence Theorem 1 continues to hold, i.e., the system is stable in the mean.*

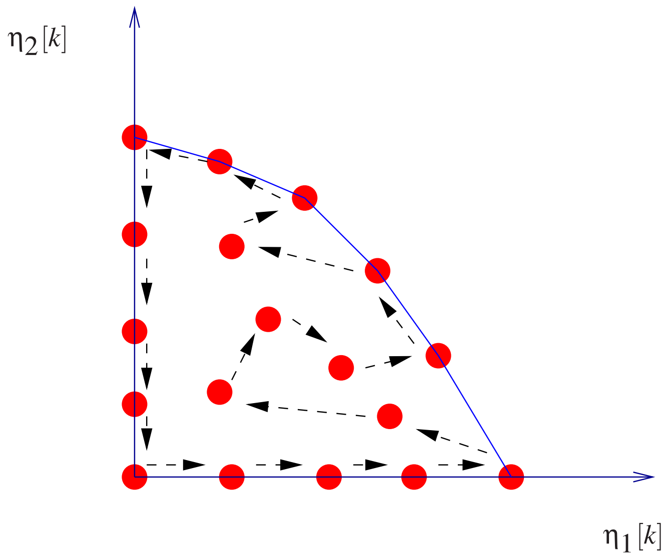


Figure 4: Finite set of rates and Hamiltonian walk

- The algorithm can be slightly modified to cover time varying channels.
- Theorem 3 will also hold for the randomized version of the algorithm.

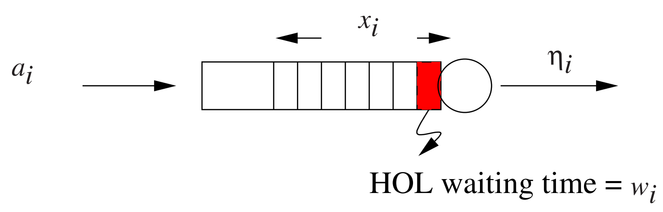
Waiting times

- Instead of the current queue length information, the controller may alternatively use the delay experienced by the packets within the queues as its input.
- Let $w_i[k]$ denote the waiting time of the head of the line (H.O.L.) packet in the i^{th} queue at time k.
- Consider a policy that chooses the service rate vector $\vec{\mu}[k]$, such that

$$\vec{\mu}[k] \in \arg \max_{\vec{\eta}[k] \in \mathcal{S}_{s[k]}} \sum_{i=1}^{N} h_i(w_i[k])\eta_i[k], \tag{5}$$

  for continuous, non-decreasing functions $\{h_i(\cdot)\}$, satisfying $h_i(0) = 0$ and $\lim_{x\to\infty} h_i(x) = \infty$.
- Under further assumptions that restrict the peakiness of the arrival process, the system is still stable-in-the-mean.

Waiting times [Cont'd]



- Mean arrival rate is λ_i.
- For large x_i,

$$x_i \approx \lambda_i w_i.$$

- $$h_i(w_i) \approx h_i(x_i/\lambda_i) =: f_i(x_i).$$

Conclusions

- Presented conditions on scheduling policies that guarantee stability for a large class of arrival and channel models.
- Analyzed several applications that incorporate

$\longrightarrow$ imperfect queue length information

$\longrightarrow$ reduction of computational complexity

$\longrightarrow$ usage of waiting times as the parameter

- Decentralized stabilizing algorithms for a network? Congestion control?

Thank you!