Imperfect Randomized Algorithms for Optimal Control of Wireless Networks

Atilla Eryilmaz
(Ohio State University)

Joint work with:
Asu Ozdaglar, Devavrat Shah, Eytan Modiano
(Massachusetts Institute of Technology)

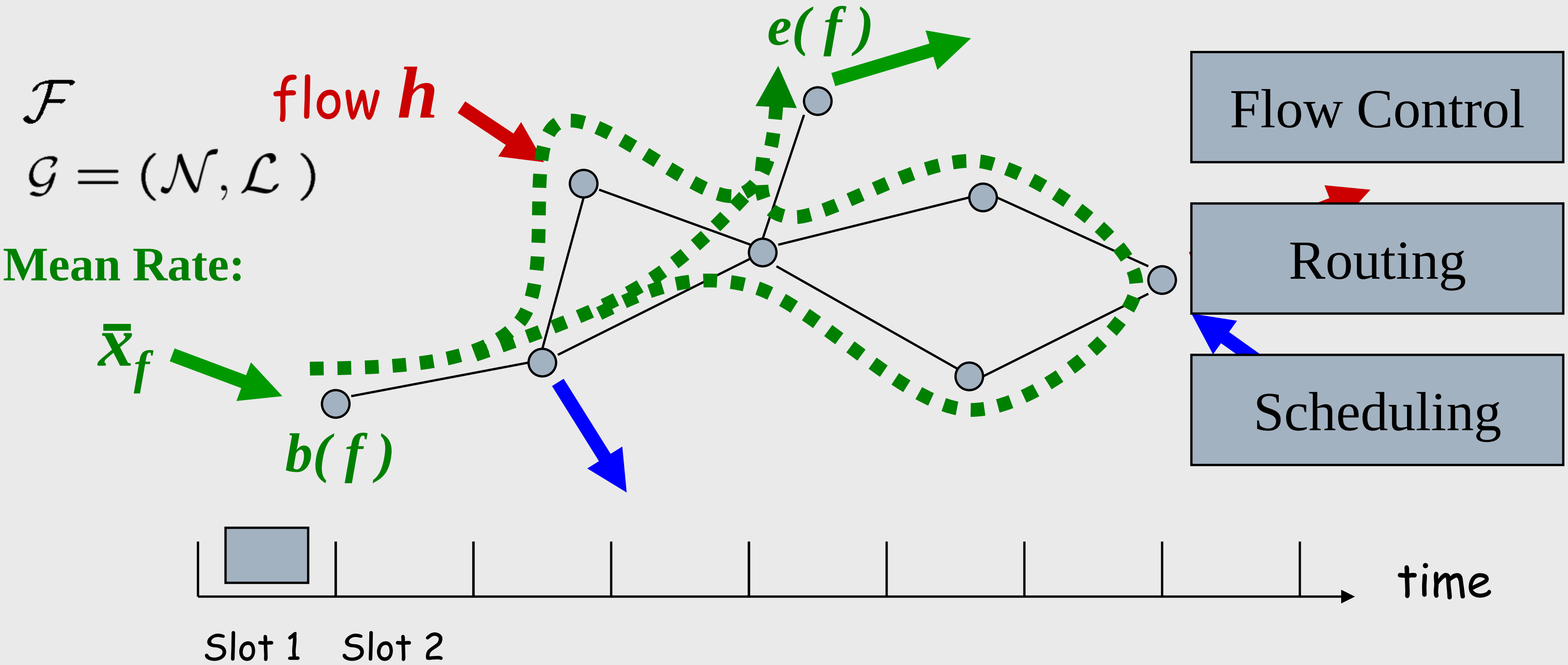
Outline

- Wireless Network Model
- Part 1 – Optimization-based Network Control
  - (Dynamic) Network Optimal Control (NOC) Problem
  - (Static) Network Utility Maximization (NUM) Problem
  - Optimal solution for NOC through Dual-NUM
  - Problem – Assumes high-complexity, centralized computations
- Part 2 – Impact of Randomized Implementations
  - Description of a class of randomized algorithms amenable to low-complexity, distributed implementation
  - Optimality characteristics under randomized implementation
- Summary & Conclusions

Outline

- Wireless Network Model
- Part 1 – Optimization-based Network Control
  - (Dynamic) Network Optimal Control (NOC) Problem
  - (Static) Network Utility Maximization (NUM) Problem
  - Optimal solution for NOC through Dual-NUM
  - Problem – Assumes high-complexity, centralized computations
- Part 2 – Impact of Randomized Implementations
  - Description of a class of randomized algorithms amenable to low-complexity, distributed implementation
  - Optimality characteristics under randomized implementation
- Summary & Conclusions

Wireless Network Model

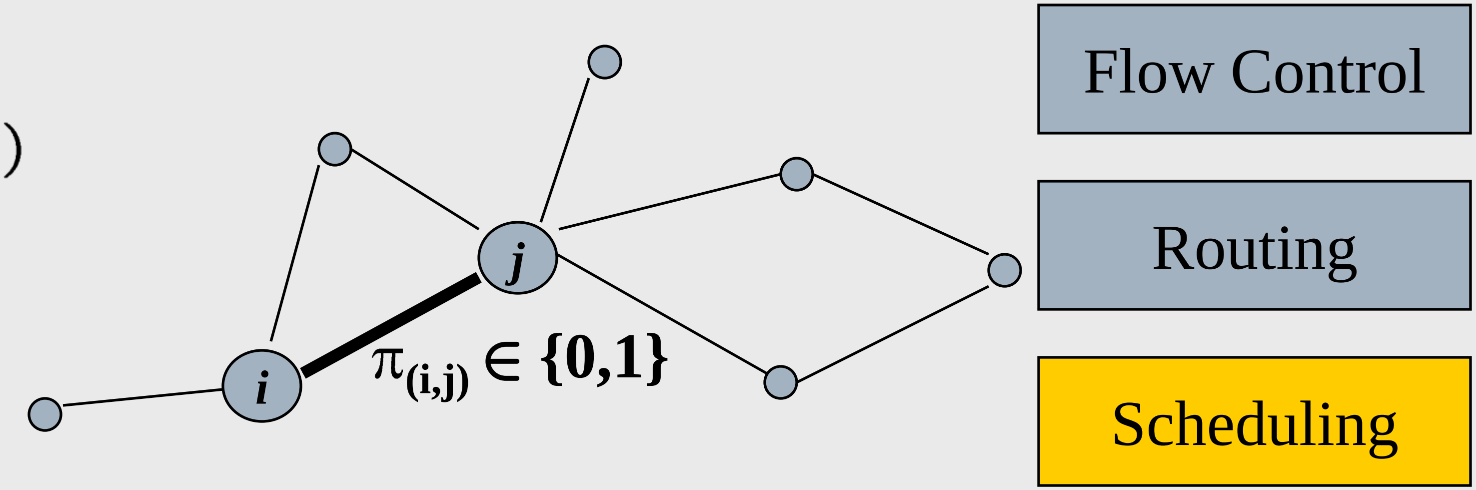


□ $U_f(\cdot)$ is a strictly concave, non-decreasing function that measures the utility of Flow-f as a function of its mean rate

Wireless Network Model

$\mathcal{F}$

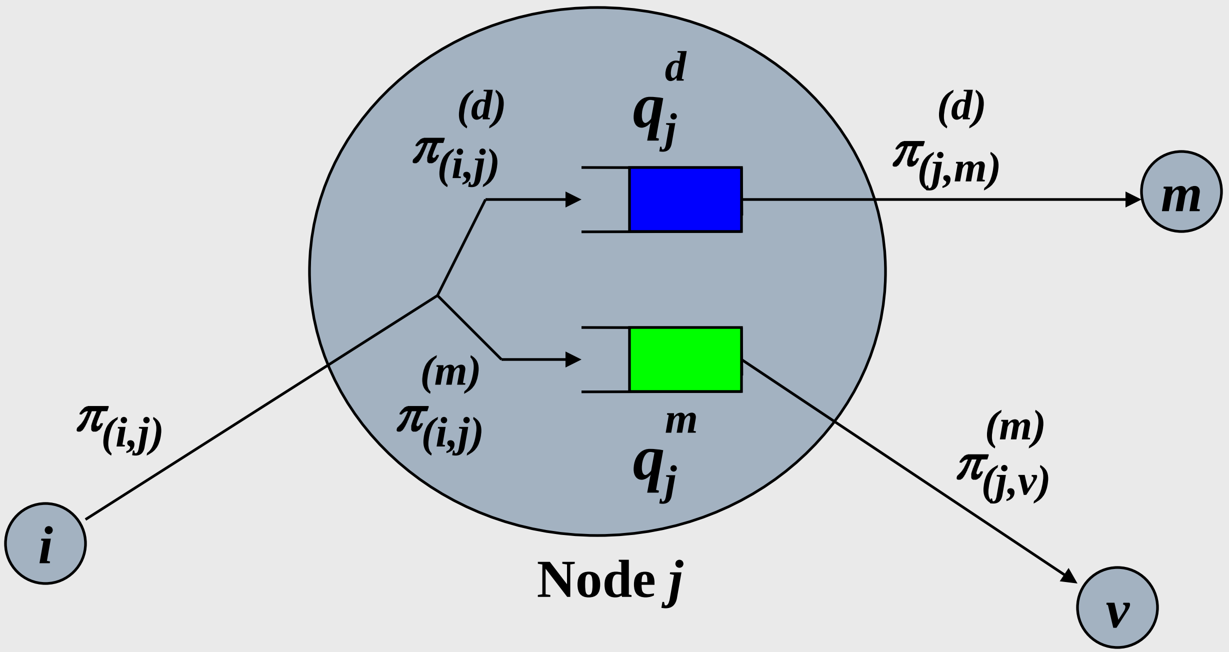
$\mathcal{G} = (\mathcal{N}, \mathcal{L})$



- $\mathcal{S}$: Set of *feasible link activation vectors* (or *feasible schedules*)
- Schedule of slot t, denoted $\boldsymbol{\pi}[t] = (\pi_{(i,j)}[t])_{(i,j)\in\mathcal{L}}$, must be in $\mathcal{S}$, $\forall t$
- Π = Convex Hull{ $\mathcal{S}$ }: *Achievable mean link rates*
- A *scheduling policy P* is a mapping from the current "state" of the system to feasible schedules
- Let $\mathcal{P}$ denote the *set of all scheduling policies*

Queueing Architecture and Evolution

- Each node maintains a queue for each destination node.



- The evolution of a queue length is described by

$$q_i^d[t+1] = \left[q_i^d[t] + x_{into(i)}^{(d)}[t] + \pi_{into(i)}^{(d)}[t] - \pi_{out(i)}^{(d)}[t]\right]^+$$

Definitions

- A queue, say q_i^d, is *stable* if

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} q_i^d[t] < \infty$$

- A *queue-length based flow control policy* $X : \mathbf{q} \rightarrow [0, M]^{|\mathcal{F}|}$ is a mapping from queue-lengths to feasible rates
- Let $\mathcal{X}$ denote the set of all queue-length-based flow control policies
- Then, the queue-length evolution for a given scheduling policy P, can be written as

$$\mathbf{q}[t+1] = f(\mathbf{q}[t], P, X(\mathbf{q}[t])),$$

for some function f

Outline

- Wireless Network Model
- Part 1 – Optimization-based Network Control
  - (Dynamic) Network Optimal Control (NOC) Problem
  - (Static) Network Utility Maximization (NUM) Problem
  - Optimal solution for NOC through Dual-NUM
  - Problem – Assumes high-complexity, centralized computations
- Part 2 – Impact of Randomized Implementations
  - Description of a class of randomized algorithms amenable to low-complexity, distributed implementation
  - Optimality characteristics under randomized implementation
- Summary & Conclusions

Network Optimal Control (NOC) Problem

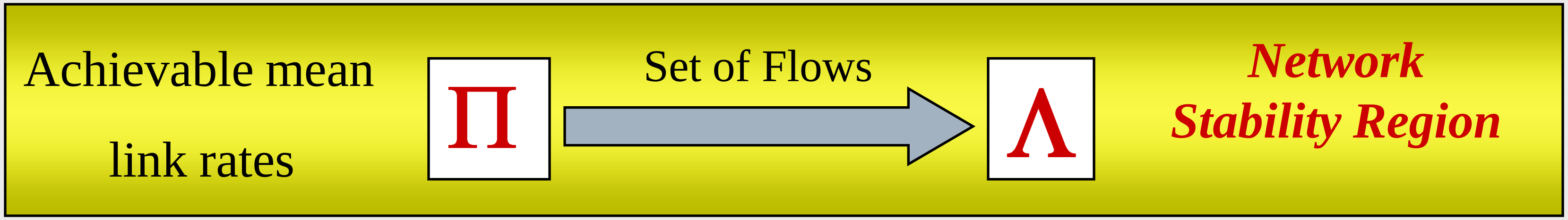
- *Network Optimal Control (NOC)* Problem:

$$\max_{X \in \mathcal{X}, P \in \mathcal{P}} \quad \sum_{f \in \mathcal{F}} U_f(\bar{x}_f)$$

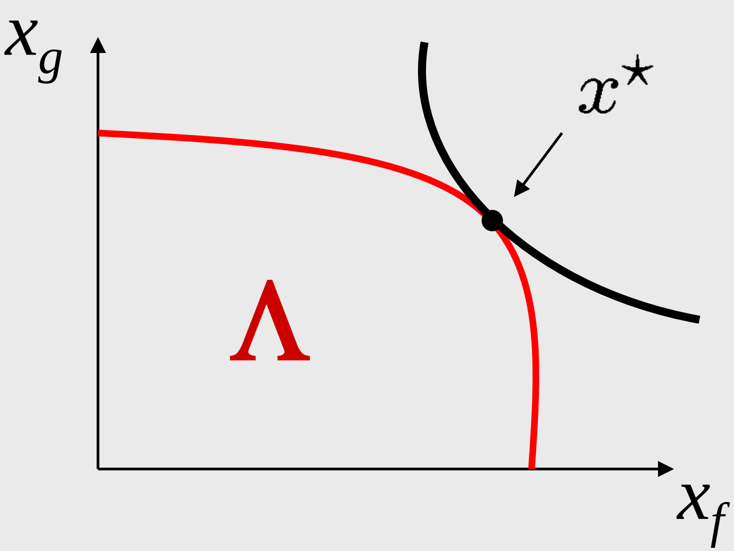
$$\begin{aligned} s.t. \quad & \mathbf{q}[0] \equiv \mathbf{0}, \qquad t \in \mathcal{Z}_+, \\ & \mathbf{q}[\mathbf{t}+1] = f\left(\mathbf{q}[\mathbf{t}], P, X(\mathbf{q}[\mathbf{t}])\right), \\ & \limsup_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} q_i^d[t] < \infty, \qquad \forall i, d \in \mathcal{N}, \end{aligned}$$

where $\bar{x}_f := \liminf_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} x_f[t].$

Network Utility Maximization (NUM) Problem



- Define the *optimal mean rate vector* ***x^**** as



$$x^\star \in \arg\max_{x} \sum_{f \in \mathcal{F}} U_f(x_f)$$
$$s.t. \quad x \in \boldsymbol{\Lambda}$$

Network Utility Maximization

(NUM)

Dual Formulation of NUM

- A ***Dual function*** associated with the previous problem is

$$D(\lambda) = \sum_{f \in \mathcal{F}} \max_{x_f \geq 0} \{U_f(x_f) - x_f \lambda_{b(f)}^{e(f)}\}$$

Distributed Flow Control

$$+ \max_{\pi \in \Pi} \sum_{(i,j) \in \mathcal{L}} \pi_{(i,j)} \max_{d \in \mathcal{N}} \left\{ \left| \lambda_i^d - \lambda_j^d \right| \right\}$$

Backpressure Scheduler/Router

where we λ_i^d can be interpreted as the price associated with sending a unit rate of flow from node *i* to node *d*.

- Then the ***Dual Problem*** is given as: $\min_{\lambda \geq \mathbf{0}} D(\lambda)$
- **Fact**: There is no duality gap, i.e., there exists a nonempty set Ψ^* such that:

$$\sum_{f \in \mathcal{F}} U_f(x_f^\star) = D(\lambda^\star), \qquad \forall \lambda^\star \in \mathbf{\Psi}^\star$$

Sub-gradient Methods to Solve NUM

- Employ Dual (or Primal-Dual) Methods:

$$\lambda_i^d[t+1] = \left[\lambda_i^d[t] + \theta_t \left(x_{into(i)}^{(d)}[t] + \pi_{into(i)}^{(d)}[t] - \pi_{out(i)}^{(d)}[t] \right)\right]^+$$

$$x_f[t] = \left[U_f'^{-1}\left(\lambda_f[t]\right)\right]_0^M$$

where θ_t is the step-size, and $\lambda_f = \lambda_{b(f)}^{e(f)}$

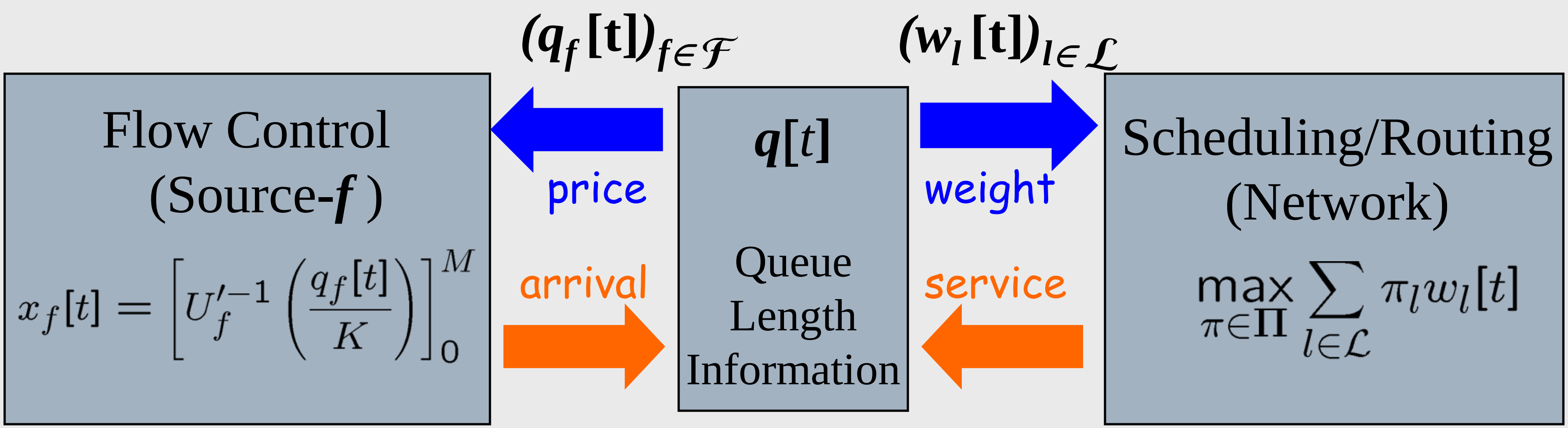
- Then, using results from optimization theory, we have, under appropriate step-size rules,

$$\lambda \rightarrow \Psi^*, \text{ and } x \rightarrow x^*$$

Cross-layer Mechanism for NOC

- **q**[t] in the network can be interpreted as a scaled version of the prices **λ**[t]
- Introduce a design parameter *K*

$$w_{(i,j)}[t] := \max_{d \in \mathcal{N}} \left| q_i^d[t] - q_j^d[t] \right|$$



- **q**[t] ≈ K **λ**[t], and the mechanism solves NOC

Relevant Literature [partial list] & Complexity Issue

- Routing/Scheduling – [Tassiulas, Ephremides ’92], [Neely, Modiano, Rohrs ’03], [Eryilmaz, Srikant ’03], [Ho, Viswanathan ’06], [Eryilmaz, D. Lun ’07]
- Optimization – [Kelly, Moullo, Tan ’98], [Low, Lapsley ’99], [Srikant ’04]
- Cross-Layer – [Stolyar ’04], [Lin, Shroff ’04], [Eryilmaz, Srikant ’05, ‘06], [Neely, Modiano ’05], [Chen, Low, Chiang, and Doyle ’06]
- The solution to $\max_{\pi \in \Pi} \sum_{l \in \mathcal{L}} \pi_l w_l[t]$, assumed in these works, is generally difficult to compute (even NP-hard for many interference models)
- Thus, the cross-layer mechanism is impractical

Outline

- Wireless Network Model
- Part 1 – Optimization-based Network Control
  - (Dynamic) Network Optimal Control (NOC) Problem
  - (Static) Network Utility Maximization (NUM) Problem
  - Optimal solution for NOC through Dual-NUM
  - Problem – Assumes high-complexity, centralized computations
- Part 2 – Impact of Randomized Implementations
  - Description of a class of randomized algorithms amenable to low-complexity, distributed implementation
  - Optimality characteristics under randomized implementation
- Summary & Conclusions

Towards Distributed Implementation

- Several low-complexity schedulers are proposed to provide approximate solutions to

$$\max_{\pi \in \mathbf{\Pi}} \sum_{l \in \mathcal{L}} \pi_l w_l[t]$$

(e.g. [Lin, Shroff '05], [Wu, Srikant '06], [Bui, Eryilmaz, Srikant, Wu '06], [Gupta, Lin, Srikant '06], [Modiano, Shah, Zussman '06], [Eryilmaz, Ozdaglar, Modiano '07], [Sanghavi, Bui, Srikant '07])

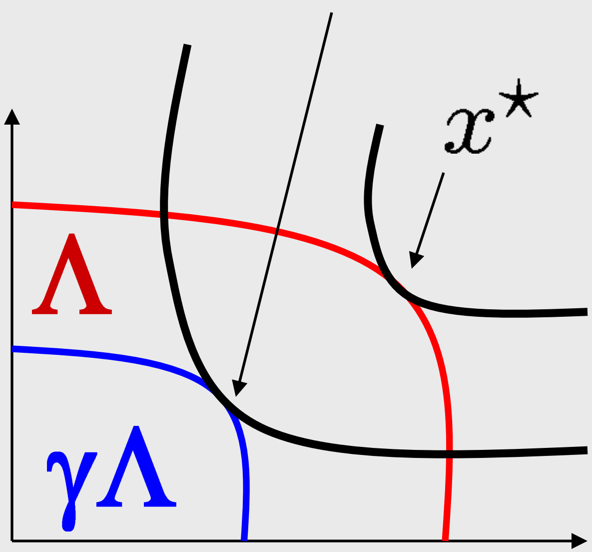
- However, little is done in understanding the optimality properties
- Our contribution: Study the optimality performance of **a large class of high-performance randomized policies** that are amenable to **low-complexity** and **distributed** implementation

An Earlier Work

- *γ-Imperfect Scheduler* [Lin, Shroff `05]: Assume that the scheduler picks π[t] such that

$$\sum_{l \in \mathcal{L}} \pi_l[t] w_l[t] \geq \gamma \max_{\pi \in \Pi} \sum_{l \in \mathcal{L}} \pi_l w_l[t], \text{ for some } \gamma \in (0,1] \quad \textbf{(1)}$$

$$x^\star(\gamma) = \arg\max_{x \in \gamma\Lambda} \sum_f U_f(x_f)$$



For a given ε > 0,

$$\sum_f U_f(\bar{x}_f) \geq \sum_f U_f(x_f^\star(\gamma)) - \epsilon,$$

where $\bar{x}_f := \liminf_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} x_f[t]$

Comments

- Greedy Maximal Schedulers are available that satisfy **(1)** with $\gamma \leq$ ½
- However, these schedulers can perform arbitrarily badly depending on the interference model [Chaporkar, Kar, Sarkar ‘05]
- It is difficult to find low-complexity algorithms that guarantee **(1)** with γ close to 1.
- We study the network utilization factor of a **generic class of randomized schedulers that iteratively improve the schedule as the system evolves**

Joint Scheduling-Routing-Flow Control Policy

1. Dual Flow Control Policy:

- At each slot t, Flow-f updates its arrival rate as

$$x_f[t] \;=\; \left[U_f'^{-1}\left(\frac{q_f[t]}{K}\right)\right]_0^M,$$

where K and M are positive design parameters.

- K can be interpreted as a measure of aggressiveness of the flow controller.

Joint Scheduling-Routing-Flow Control Policy

2. Generic Randomized Scheduling-Routing Policy:

- Define link weights: $$w_{(i,j)}[t] = \max_d \left| q_i^d[t] - q_j^d[t] \right|$$
- Let $$\pi^*[t] \in \arg\max_{\pi\in\Pi} (\pi \cdot \mathbf{w}[t])$$ $$\pi^*[t] \in \arg\max_{\pi\in\Pi} \sum_{(n,m)\in\mathcal{L}} \pi_{(n,m)} w_{(n,m)}[t]$$
- There exists a randomized policy **(R)** that picks a schedule $\pi^{(R)}$ satisfying

$$P\left(\pi^{(R)} = \pi^*[t] \mid \mathbf{q[t]}\right) \geq \delta > 0 \qquad \forall \mathbf{q[t]}.$$

- Repeat:

PICK

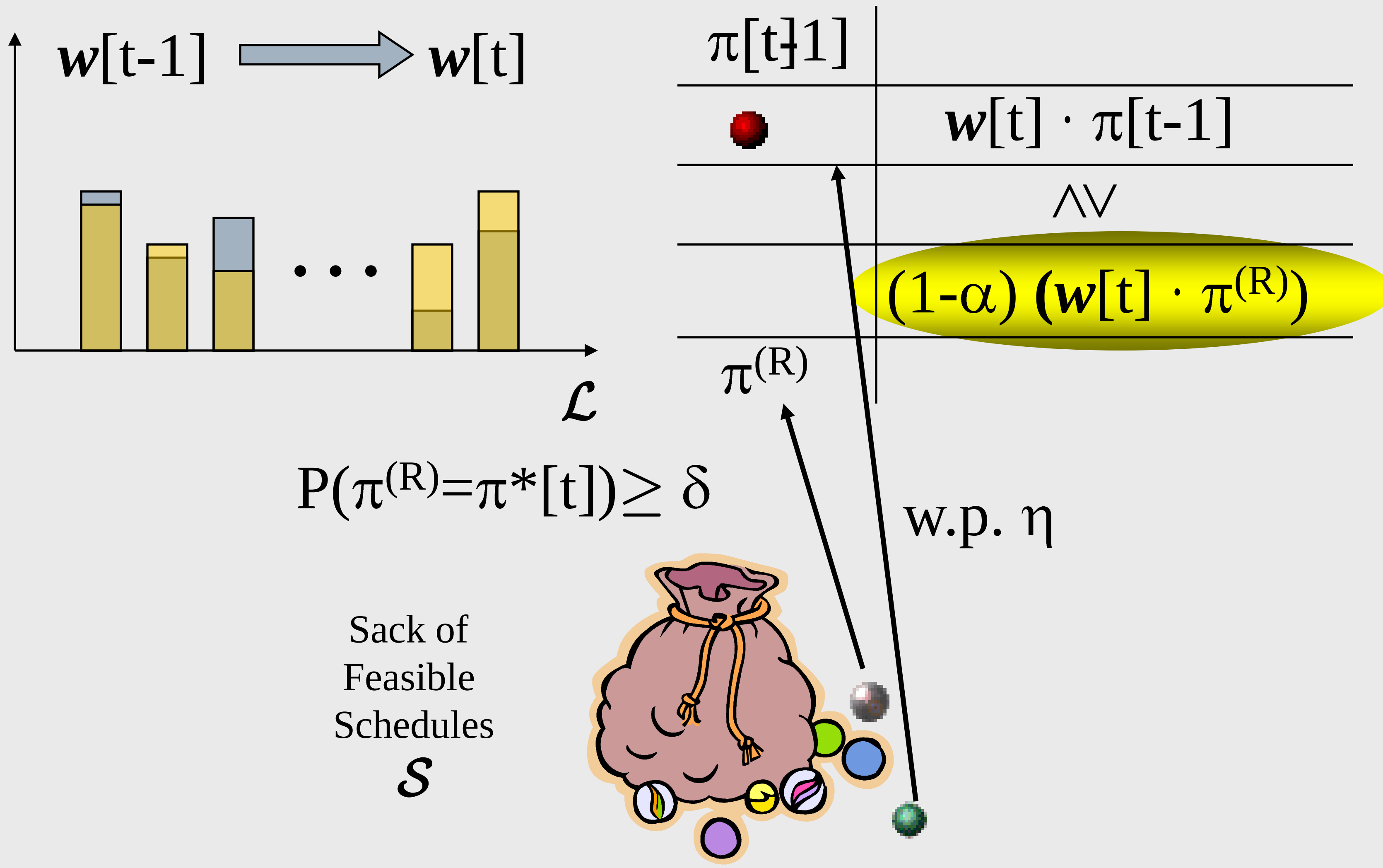
$$\pi^{(R)}[t] \leftarrow \text{Pick a random allocation;}$$

$$\text{Set } \pi[t] \quad s.t. \quad P\Big((\mathbf{w}[t]\cdot\pi[t]) \geq \max\{(\mathbf{w}[t]\cdot\pi[t-1]), (1-\alpha)(\mathbf{w}[t]\cdot\pi^{(R)}[t])\}\Big) \geq (1-\eta)$$

$$t \leftarrow t+1;$$

COMPUTE **COMPARE**

Visualization – Picking π[t]



Main Result

- Under the Generic Randomized Cross-layer Controller, we have

$$\sum_f U_f(\bar{x}_f) \geq \sum_f U_f\left(x_f^\star\left(1-\alpha-2\sqrt{\frac{\eta}{\delta}}\right)\right) - \frac{M^2|\mathcal{L}|}{2K},$$

where $\bar{x}_f := \liminf_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} x_f[t]$

- Shows that the network utilization factor γ satisfies

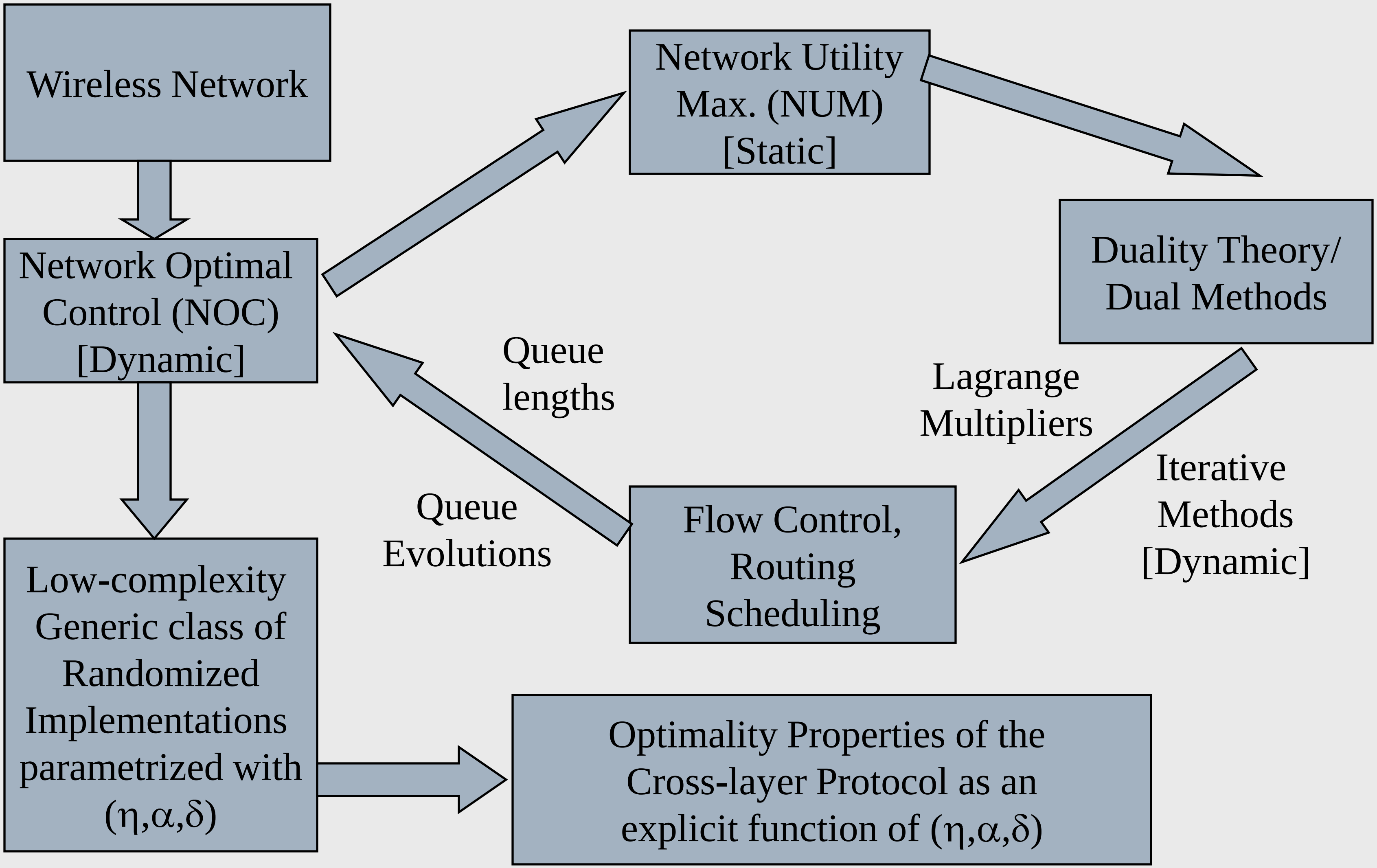
$$\gamma = 1-\alpha-2\sqrt{\frac{\eta}{\delta}}$$

- Easy to design low-complexity schedulers with $\gamma \approx 1$

Example Algorithms

- [Modiano, Shah, Zussman ’06] –
  - First-order interference model with $\eta=\alpha=0$, $\delta>0$
- [Eryilmaz, Ozdaglar, Modiano ’07] –
  - General interference model with $\eta=\alpha=0$, $\delta>0$
- [Modiano, Shah, Zussman ’06] – Gossip Algorithms
  - η and α can be set to arbitrarily small values at the expense of slower convergence
- [Sanghavi, Bui, Srikant `07] –
  - First-order interference model with $\eta=0$, $\alpha=1/m$, $\delta>0$ for a design parameter m

Summary – A roadmap



Conclusions & Future Directions

- Identified the relationship between the degree of optimality and the algorithm parameters
- Observed that high degree of optimality is achievable with these schedulers with low-complexity implementations
- Development of other schedulers with favorable qualities in terms of metrics of interest (e.g. complexity, delay, rate of convergence) that fit into this framework

Thank you !

Questions?

Supplemental Slides

Translation of Π to Λ

- $\boldsymbol{x} = (x_{b(f)}^{e(f)})_f \geq \mathbf{0} \in \Lambda$ if
  - there exists a $\pi \in \Pi$ for which we have:

$$x_i^d + \pi_{into(i)}^d \leq \pi_{out(i)}^d, \qquad d, i \in \mathcal{N}$$

(Flow Conservation Constraints)