Stochastic Network Utility Maximization

6.263/16.37 - Data Communication Networks

Igor Kadota

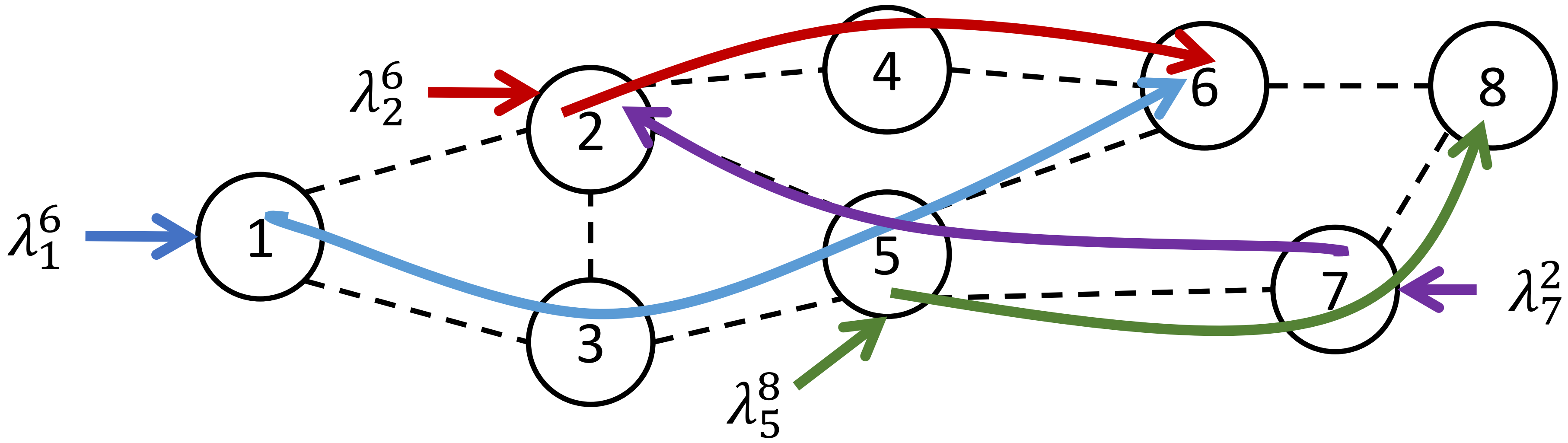
Cambridge, December 07, 2018

Outline

- Multi-commodity flow problem
- Recap from previous lectures
  - Network of Queues
  - Capacity Region and Stationary Randomized Policy
- Overloaded system
  - Problem Statement
  - Utility Function
  - Drift-Plus Penalty Algorithm (Admission + Routing + Scheduling)
  - Performance Analysis and Optimality Results

*slides adapted from Chih-Ping Li's lecture, with additional material from "Resource Allocation and Cross-Layer Control in Wireless Networks" by L. Georgiadis, M. J. Neely and L. Tassiulas and from M. J. Neely's PhD thesis.

Multi-Commodity Flow Problem



- Discrete-time system with slot index t. Network with N nodes and K commodities.
- Each node $n \in \{1,2,\ldots,N\}$ keeps per-commodity queues;
- Commodity-c packets are addressed to node c;
- λ_n^c time-average arrival rate of commodity-c packets at node n;
- **Goal is to find a network control algorithm that supports the arrivals λ_n^c (when possible).**

Network of Queues

- Let $Q_n^c(t)$ be the number of commodity-c packets enqueued at node n at the beginning of time-slot t. Then, according to Lindley recursion:

$$Q_n^c(t+1) \leq \max\left\{Q_n^c(t) - \sum_{j=1}^{N} \mu_{n,j}^c(t)\,;0\right\} + \sum_{i=1}^{N} \mu_{i,n}^c(t) + A_n^c(t)$$

where:

- $A_n^c(t) \geq 0$ is the number of **exogenous** packet arrivals at the end of slot t;
- $\mu_{n,j}^c(t) \geq 0$ is the **offered** transmission opportunity to comm.-c over (n,j) during slot t;
- $\mu_{n,j}(t) = \sum_{c=1}^{K} \mu_{n,j}^c(t)$ is the total transmission opportunity over (n,j) during slot t;
- $0 \leq f_{n,j}^c(t) \leq \mu_{n,j}^c(t)$ is the # of packet **transmissions** of comm.-c over (n,j) during slot t;
- $\mathcal{C}_{n,j}$ is the **capacity** constraint of link (n,j).

Network of Queues

- Let $Q_n^c(t)$ be the number of commodity-c packets enqueued at node n at the beginning of time-slot t. Then, according to Lindley recursion:

$$Q_n^c(t+1) = Q_n^c(t) - \sum_{j=1}^{N} f_{n,j}^c(t) + \sum_{i=1}^{N} f_{i,n}^c(t) + A_n^c(t)$$

where:

- $A_n^c(t) \geq 0$ is the number of **exogenous** packet arrivals at the end of slot t;
- $\mu_{n,j}^c(t) \geq 0$ is the **offered** transmission opportunity to comm.-c over (n, j) during slot t;
- $\mu_{n,j}(t) = \sum_{c=1}^{K} \mu_{n,j}^c(t)$ is the total transmission opportunity over (n, j) during slot t;
- $0 \leq f_{n,j}^c(t) \leq \mu_{n,j}^c(t)$ is the # of packet **transmissions** of comm.-c over (n, j) during slot t;
- $\mathcal{C}_{n,j}$ is the **capacity** constraint of link (n, j).

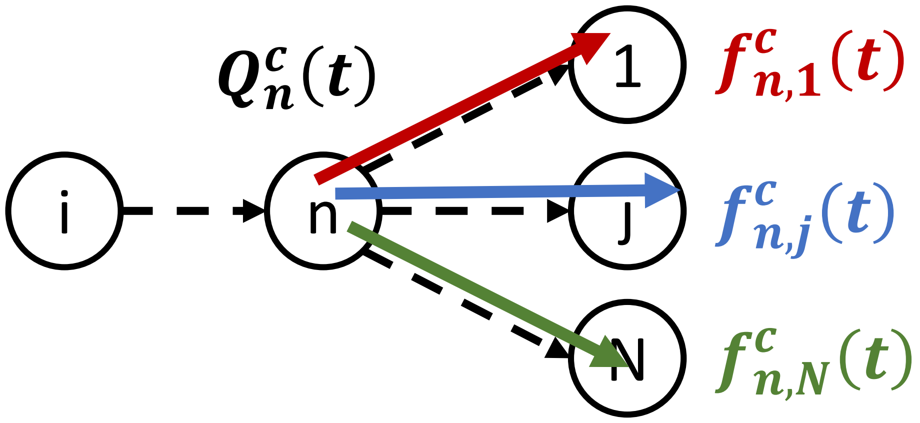
Network of Queues

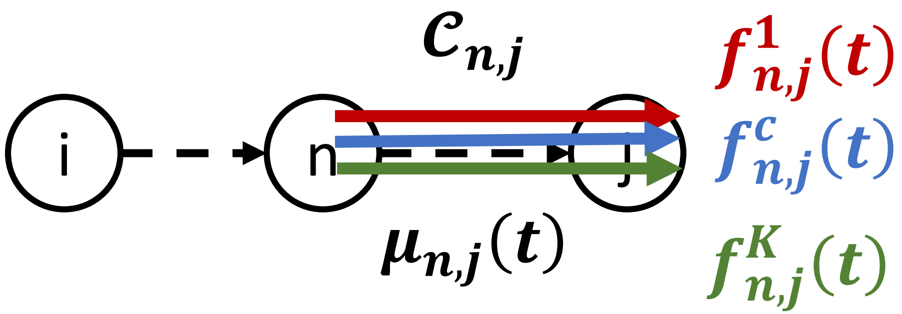
- Let $Q_n^c(t)$ be the number of commodity-c packets enqueued at node n at the beginning of time-slot t. Then, according to Lindley recursion:

$$Q_n^c(t+1) = Q_n^c(t) - \sum_{j=1}^{N} f_{n,j}^c(t) + \sum_{i=1}^{N} f_{i,n}^c(t) + A_n^c(t)$$

Constraints:

$$\sum_{j=1}^{N} f_{n,j}^c(t) \le Q_n^c(t), \forall n, c, t \qquad \sum_{c=1}^{K} f_{n,j}^c(t) \le \mu_{n,j}(t) \le \mathcal{C}_{n,j}, \forall n, j, t$$





Network of Queues

- Let $Q_n^c(t)$ be the number of commodity-c packets enqueued at node n at the beginning of time-slot t. Then, according to Lindley recursion:

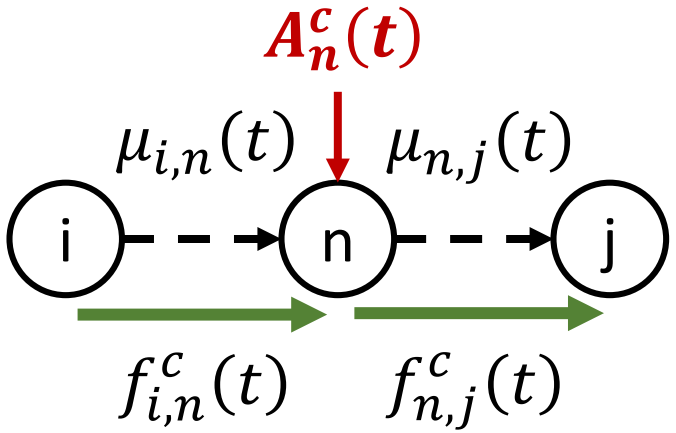
$$Q_n^c(t+1) = Q_n^c(t) - \sum_{j=1}^{N} f_{n,j}^c(t) + \sum_{i=1}^{N} f_{i,n}^c(t) + A_n^c(t)$$

Assumptions:

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} A_n^c(\tau) = \lambda_n^c \quad \text{w.p.}\,1$$

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} f_{n,j}^c(\tau) = \bar{f}_{n,j}^c \quad \text{w.p.}\,1$$

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mu_{n,j}^c(\tau) = \bar{\mu}_{n,j}^c \quad \text{w.p.}\,1$$



Network of Queues – Stability is the objective

- **Rate stability**: $\lim_{t\to\infty} \frac{Q_n^c(t)}{t} = 0 \quad \text{w.p.}\,1$

Arrival Rate = Departure ≤ Dep. Opport.

  - Queue $Q_n^c(t)$ is rate stable if and only if $\sum_{i=1}^{N} \bar{f}_{i,n}^c + \lambda_n^c = \sum_{j=1}^{N} \bar{f}_{n,j}^c \le \sum_{j=1}^{N} \bar{\mu}_{n,j}^c$
  - If *Arrival* > *Departure*, then: $\lim_{t\to\infty} \frac{Q_n^c(t)}{t} = \sum_{i=1}^{N} \bar{f}_{i,n}^c + \lambda_n^c - \sum_{j=1}^{N} \bar{f}_{n,j}^c \quad \text{w.p.}\,1$

- **Strong stability:** $\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}[Q_n^c(\tau)] < \infty$ [bounded time-average queue backlog]
  - If $Q_n^c(t)$ is strongly stable and there exists $C > 0$ such that

$$arr(t) - dep\_opp(t) \le C \quad \text{w.p.}\,1, \forall t \quad \text{or} \quad dep\_opp(t) - arr(t) \le C \quad \text{w.p.}\,1, \forall t$$

Then $Q_n^c(t)$ is also Rate Stable.

Network of Queues – Action and Outcome

- **Network State**: $s(t) \in \mathcal{S} \leftarrow$ channels, topology, packet arrivals,… [uncontrollable]
  - Assumption: s(t) evolves according to an irreducible MC with finite states such that :

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathfrak{I}ndicator_{\{s(\tau)=s\}} = \pi_s \quad \text{w.p.1.} \ ; \ \forall s \in \mathcal{S}$$

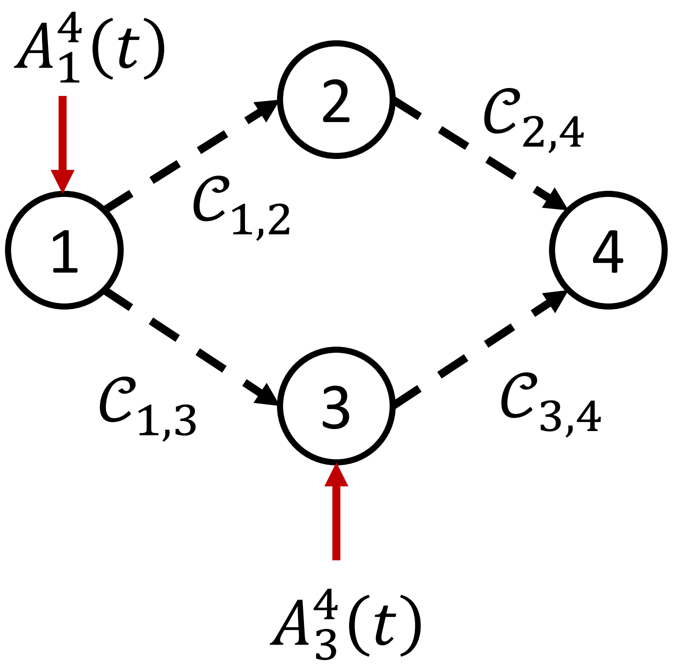
- **Action**: $I(t) \in \mathcal{I}_{s(t)} \leftarrow$ controls $\mu_{i,j}(t)$ and $f_{i,j}^{c}(t)$ using knowledge of the network state and (possibly) other information such as current queue backlog. $I(t)$ satisfies constraints from the state space $\mathcal{I}_{s(t)}$.

- **Outcome**: the result of any given state action pair $(s(t), I(t))$ are: 1) $A_n^c(t)$ for every node n and; 2) $\mu_{i,j}(t)$ and $f_{i,j}^{c}(t)$ for every link (i,j) and commodity c.
  - Define the (total) link transmission rate **matrix** as $\boldsymbol{U}\big(s(t), I(t)\big) = \left[\mu_{i,j}(t)\right]_{i,j}$.

Capacity Region – Definition

Definition: Capacity Region Λ is the closure of the set of all arrival rate matrices $(\lambda_n^c)_{n,c}$ that can be stabilized by some (possibly unknown) network control algorithm.

Example: wireline network with capacity $\mathcal{C}_{n,j}$. State is $s(t) = \text{all channels are ON}, \forall t$.



The capacity region Λ is given by the set of all λ_1^4, λ_3^4 for which there exists flow variables $\bar{f}_{n,j}^c$ satisfying:

Capacity: $\sum_{c=1}^{K} \bar{f}_{i,j}^c \leq \mathcal{C}_{i,j}, \forall i, j;$

Conservation: $\lambda_n^c = \sum_{j=1}^{N} \bar{f}_{n,j}^c - \sum_{i=1}^{N} \bar{f}_{i,n}^c, \forall n, c;$

Notice similarity between flow conservation and the necessary and sufficient conditions for rate stability.

Capacity Region – Definition

Definition: Capacity Region Λ is the closure of the set of all arrival rate matrices $(\lambda_n^c)_{n,c}$ that can be stabilized by some (possibly unknown) network control algorithm.

Supported transmission rate matrices: $\mathcal{Cl}(\Gamma)$

- Consider a fixed state $s \in \mathcal{S}$ and define the set A_s of all possible transmission rate matrices:

$$A_s := \left\{ \boldsymbol{U}(s, I) = \left[\mu_{i,j}(t)\right]_{i,j} \middle| \, I \in \mathcal{I}_s \right\}$$

- Consider the Stationary Randomized policy which selects $I(t) = I$ w.p. $p_s(I) \in (0,1]$.
  - This stationary policy attains $\mathbb{E}\left[\left[\mu_{i,j}(t)\right]_{i,j} \middle| s(t) = s\right] \in ConvHull\{A_s\}$.
  - By appropriately selecting $p_s(I)$, any point in $ConvHull\{A_s\}$ can be achieved.
- The long-term link transmission rate matrix achieved by the class of Stationary Randomized policies is defined as $\Gamma := \sum_{s \in \mathcal{S}} \pi_s ConvHull\{A_s\}$ and its closure is denoted $\mathcal{Cl}(\Gamma)$.

Capacity Region – Definition

Definition: Capacity Region Λ is the closure of the set of all arrival rate matrices $(\lambda_n^c)_{n,c}$ that can be stabilized by some (possibly unknown) network control algorithm.

<u>Illustration:</u>

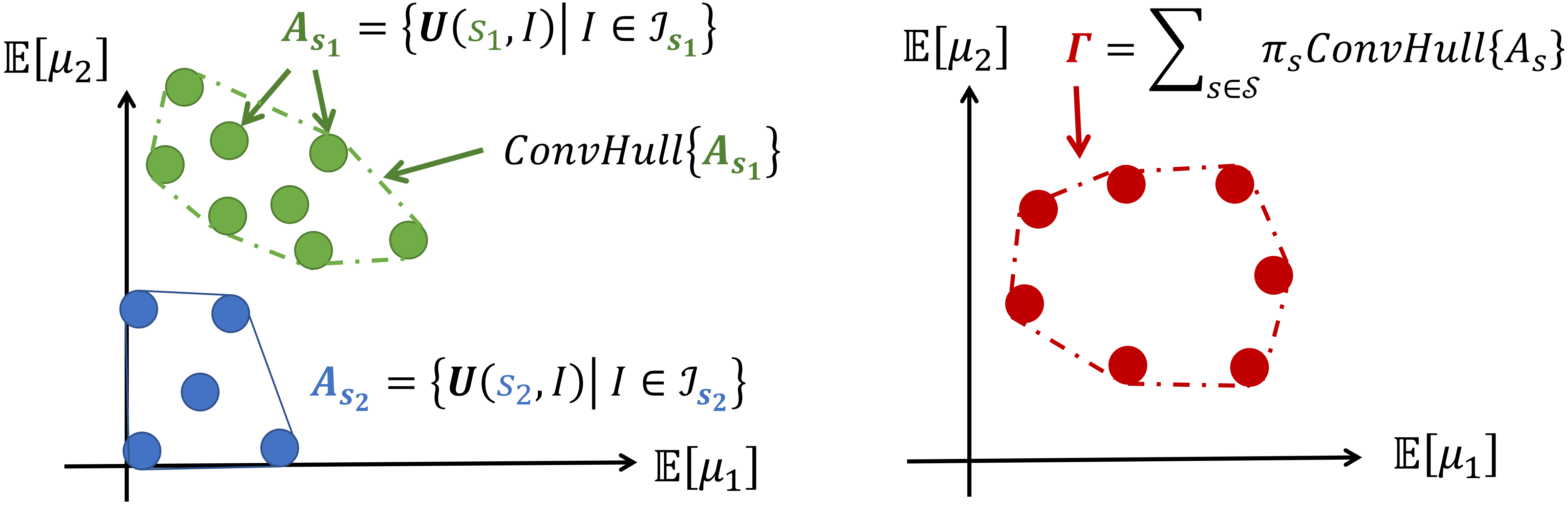


*figure adapted from the book "Resource Allocation and Cross-Layer Control in Wireless Networks"

Capacity Region – Theorem

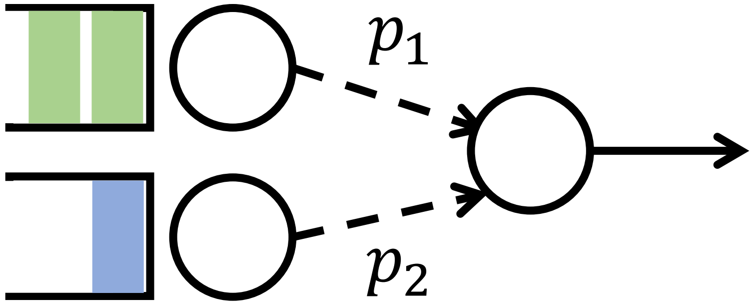
Definition: Capacity Region Λ is the closure of the set of all arrival rate matrices $(\lambda_n^c)_{n,c}$ that can be stabilized by some (possibly unknown) network control algorithm.

Theorem: the capacity region Λ is given by the set of arrival rate matrices $(\lambda_n^c)_{n,c}$ for which there exists a transmission rate matrix $\mathbb{E}\left[\mu_{i,j}(t)\right]_{i,j} = \left[\bar{\mu}_{i,j}\right]_{i,j} \in \mathcal{C}l(\Gamma)$ together with multi-commodity flow variables $\bar{f}_{i,j}^c$ that satisfy the following routing feasibility constraints:

- Flow capacity: $\sum_{c=1}^{K} \bar{f}_{i,j}^c \leq \bar{\mu}_{i,j} \leq \mathcal{C}_{i,j}, \forall i, j;$
- Flow conservation: $\lambda_n^c = \sum_{j=1}^{N} \bar{f}_{n,j}^c - \sum_{i=1}^{N} \bar{f}_{i,n}^c, \forall n, c;$
- Flow are efficient: $\bar{f}_{i,i}^c = 0, \bar{f}_{c,i}^c = 0;$ [no flow from a node to itself or from the destination]
- Flows are non-negative: $\bar{f}_{i,j}^c \geq 0, \forall i, j, c;$
- Routing constraints: $\bar{f}_{i,j}^c = 0, \forall (i, j, c)$ for which flow is not allowed.

Capacity Region – Example On/Off downlink

- i.i.d. Bernoulli channel state with $\mathbb{P}\{s_1(t) = ON\} = p_1$ and $\mathbb{P}\{s_2(t) = ON\} = p_2$
- Every time-slot t, the controller observes the channels $s_n(t)$ and serves at most one packet from one of the queues: $\mu_n(t) \in \{0,1\}$ such that $\mu_1(t) + \mu_2(t) \leq 1, \forall t$.



State	Probability	Transmission Rates
(OFF,OFF)	$(1\text{-}p_1)(1\text{-}p_2)$	(0,0)
(ON,OFF)	$p_1(1\text{-}p_2)$	(0,0), (1,0)
(OFF,ON)	$(1\text{-}p_1)p_2$	(0,0), (0,1)
(ON,ON)	p_1p_2	(0,0), (1,0), (0,1)

- Long-term link transmission rate matrix:

$$\boldsymbol{\Gamma} = (1-p_1)(1-p_2)\{(\mathbf{0},\mathbf{0})\} + p_1(1-p_2)\boldsymbol{ConvHull}\{(\mathbf{0},\mathbf{0}),(\mathbf{1},\mathbf{0})\} +$$

$$+(1-p_1)p_2\boldsymbol{ConvHull}\{(\mathbf{0},\mathbf{0}),(\mathbf{0},\mathbf{1})\} + p_1p_2\boldsymbol{ConvHull}\{(\mathbf{0},\mathbf{0}),(\mathbf{0},\mathbf{1}),(\mathbf{1},\mathbf{0})\}$$

Capacity Region – Example On/Off downlink

- i.i.d. Bernoulli channel state with $\mathbb{P}\{s_1(t) = ON\} = p_1$ and $\mathbb{P}\{s_2(t) = ON\} = p_2$
- Every time-slot t, the controller observes the channels $s_n(t)$ and serves at most one packet from one of the queues: $\mu_n(t) \in \{0,1\}$ such that $\mu_1(t) + \mu_2(t) \leq 1, \forall t$.

- Long-term link transmission rate matrix:

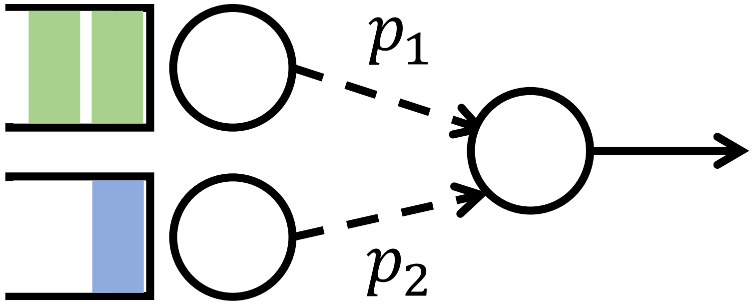
$$\boldsymbol{\Gamma} = (1-p_1)(1-p_2)\{(\mathbf{0},\mathbf{0})\} + p_1(1-p_2)\boldsymbol{ConvHull}\{(\mathbf{0},\mathbf{0}),(\mathbf{1},\mathbf{0})\} + \\ +(1-p_1)p_2\boldsymbol{ConvHull}\{(\mathbf{0},\mathbf{0}),(\mathbf{0},\mathbf{1})\} + p_1p_2\boldsymbol{ConvHull}\{(\mathbf{0},\mathbf{0}),(\mathbf{0},\mathbf{1}),(\mathbf{1},\mathbf{0})\}$$

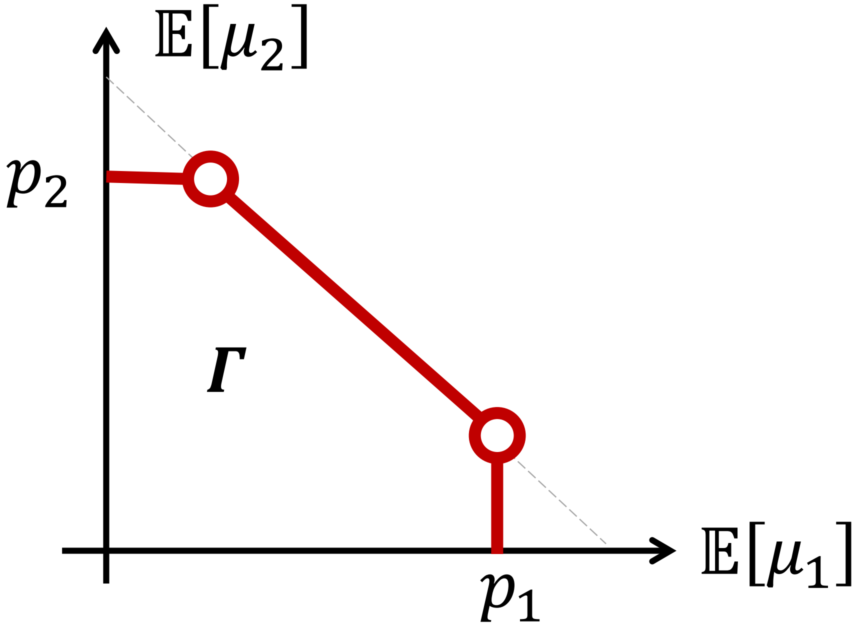
$$\boldsymbol{\Gamma} = (1-p_1)(1-p_2)(\mathbf{0},\mathbf{0}) + p_1(1-p_2)(\boldsymbol{q_1},\mathbf{0}) + (1-p_1)p_2(\mathbf{0},\boldsymbol{q_2}) + p_1p_2(\mathbf{1}-\boldsymbol{q_3},\boldsymbol{q_3})$$

$$\boldsymbol{\Gamma} = (p_1(1-p_2)\boldsymbol{q_1} + p_1p_2(\mathbf{1}-\boldsymbol{q_3}),(1-p_1)p_2\boldsymbol{q_2} + p_1p_2\boldsymbol{q_3}), \text{for } q \in [0,1]$$

Capacity Region – Example On/Off downlink

- i.i.d. Bernoulli channel state with $\mathbb{P}\{s_1(t) = ON\} = p_1$ and $\mathbb{P}\{s_2(t) = ON\} = p_2$
- Every time-slot t, the controller observes the channels $s_n(t)$ and serves at most one packet from one of the queues: $\mu_n(t) \in \{0,1\}$ such that $\mu_1(t) + \mu_2(t) \leq 1, \forall t$.



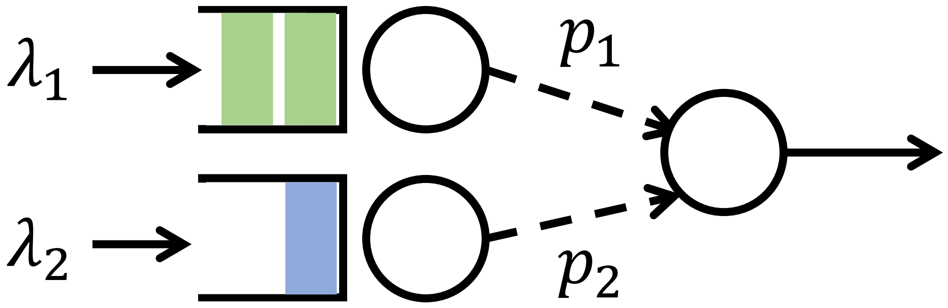


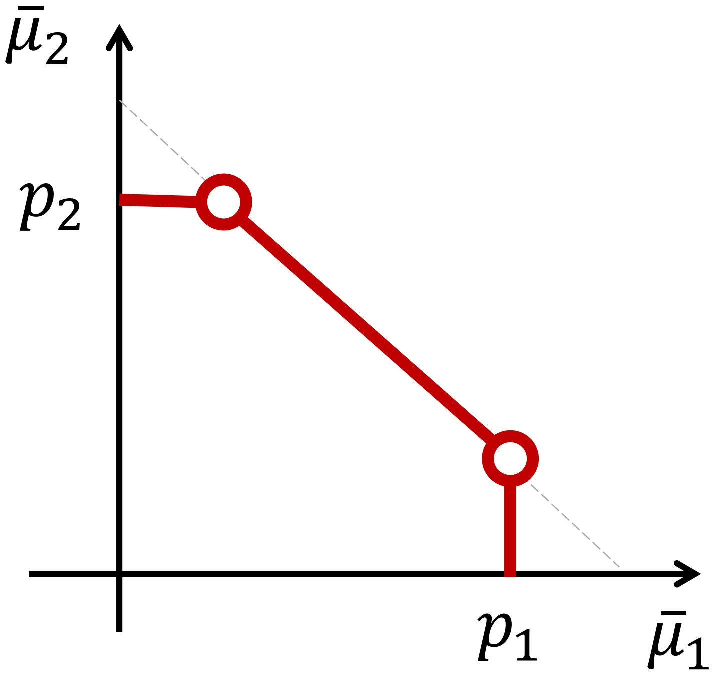
- Long-term link transmission rate matrix:

$$\boldsymbol{\Gamma} = \left(p_1(1-p_2)\boldsymbol{q_1} + p_1 p_2 (\boldsymbol{1} - \boldsymbol{q_3}),\ (1-p_1)p_2\boldsymbol{q_2} + p_1 p_2 \boldsymbol{q_3}\right)$$

Capacity Region – Example On/Off downlink

- **i.i.d. Bernoulli packet arrivals with $\mathbb{P}\{A_1(t) = 1\} = \lambda_1$ and $\mathbb{P}\{A_2(t) = 1\} = \lambda_2$.**
- i.i.d. Bernoulli channel state with $\mathbb{P}\{s_1(t) = ON\} = p_1$ and $\mathbb{P}\{s_2(t) = ON\} = p_2$
- Every time-slot t, the controller observes the channels $s_n(t)$ and serves at most one packet from one of the queues: $\mu_n(t) \in \{0,1\}$ such that $\mu_1(t) + \mu_2(t) \leq 1, \forall t$.





- **We already know the link transmission rate matrix Γ.**
- Flow conservation + Flow capacity yields:
  - $\lambda_n \leq \bar{f}_n \leq \bar{\mu}_n = \mathbb{E}[\mu_n(t)],\ n \in \{1,2\}$.
- **Conclusion: $\Gamma = \Delta$.**

Capacity Region – Randomized policy

Corollary: consider a **Stationary Randomized policy** that observes $s(t) = s$ and select a control $I(t) = I$ according to $p_s(I)$. Notice that $p_s(I)$ disregards $Q_n^c(t)$. If an arrival rate matrix $(\lambda_n^c)_{n,c}$ is interior to Λ, then *there is a randomized policy that stabilizes the system.*

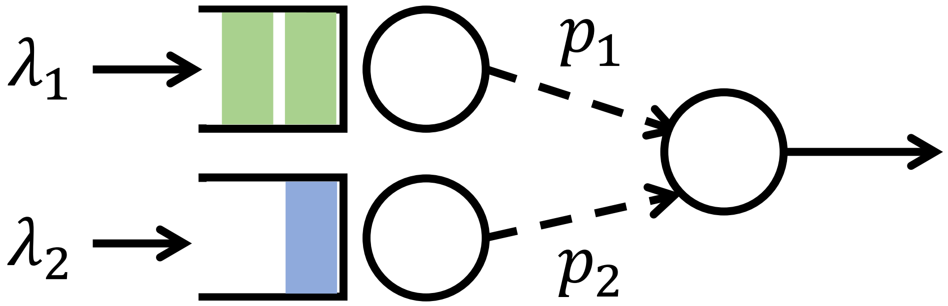
Interpretation: the randomized policy manages packet flows as a “continuous fluid”.

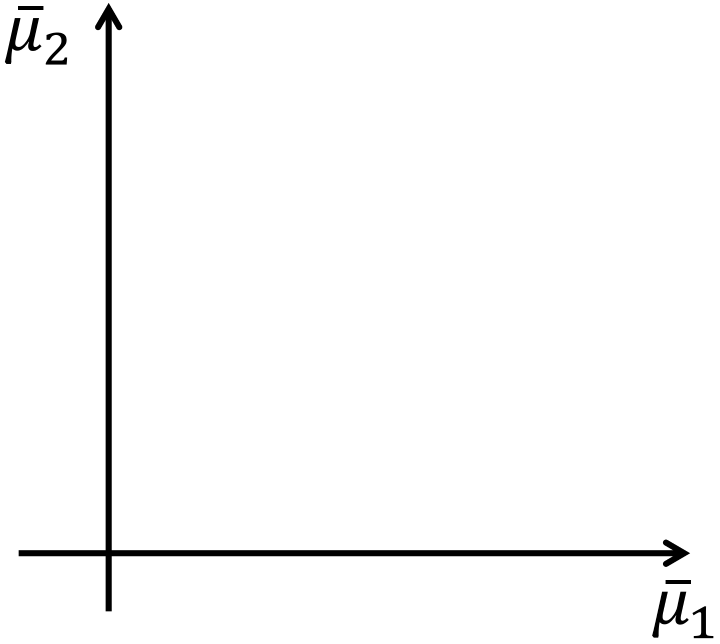
- it schedules links randomly - according to $p_s(I)$ - in order to attain the target time-average packet transmission rates $\left[\bar{\mu}_{i,j}\right]_{i,j}$.
- then, it splits the total rate $\bar{\mu}_{i,j}$ among commodities c, such that the time-average rates $\bar{\mu}_{i,j}^c$ accommodate all flows that pass through link (i,j), namely $\bar{f}_{i,j}^c \leq \bar{\mu}_{i,j}^c, \forall c$.
- this way, it can support all flows and $\lambda_n^c = \sum_{j=1}^N \bar{f}_{n,j}^c - \sum_{i=1}^N \bar{f}_{i,n}^c$

Question: is the randomized policy work-conserving? How can it be throughput optimal? What is its drawback?

Question: any work-conserving policy can stabilize the system?

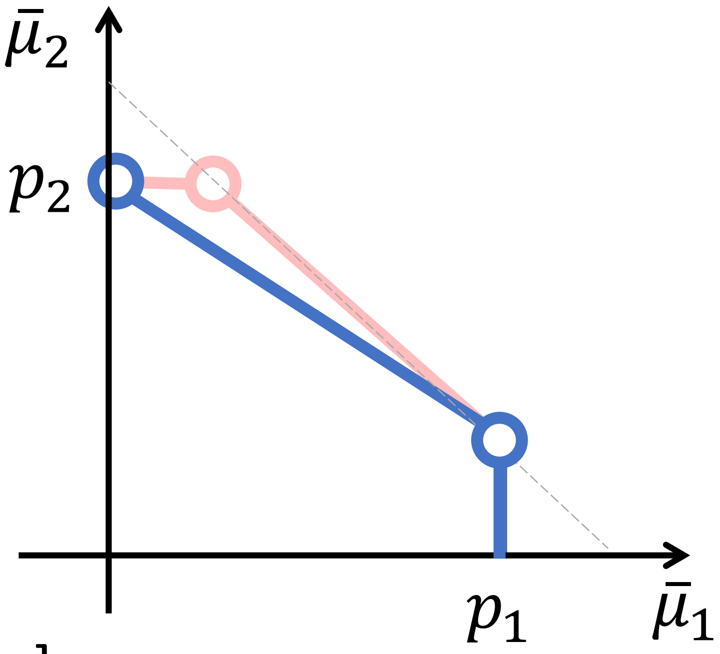
- Strict priority policy: transmits 1 while $Q_1(t) > 0$ and $s_1(t) = ON$. Transmits 2 otherwise.
- Analysis of transmission rate:
  - Queue 1 has strict priority $\rightarrow \mathbb{E}[\mu_1(t)] = \bar{\mu}_1 = ?$.
  - Queue 2 is served when Queue 1 is not served and $s_2(t) = ON \rightarrow \bar{\mu}_2 = ?$
  - Policy is throughput-optimal?





Question: any work-conserving policy can stabilize the system?

- Strict priority policy: transmits 1 while $Q_1(t) > 0$ and $s_1(t) = ON$. Transmits 2 otherwise.
- Analysis of transmission rate:
  - Queue 1 has strict priority → $\mathbb{E}[\mu_1(t)] = \bar{\mu}_1 = \min\{\lambda_1, p_1\}$.
  - Queue 2 is served when Queue 1 is not served and $s_2(t) = ON \rightarrow \bar{\mu}_2 = (1 - \bar{\mu}_1)p_2$
  - Policy is NOT throughput-optimal. See graph →



- Strict priority policy serves Queue 2 only when Queue 1 is empty. What happens if, by then, channel 2 is OFF? The transmission opportunity is lost, since $Q_1(t) = 0$. Policy does not benefit from ***multi-user diversity gain*****.**

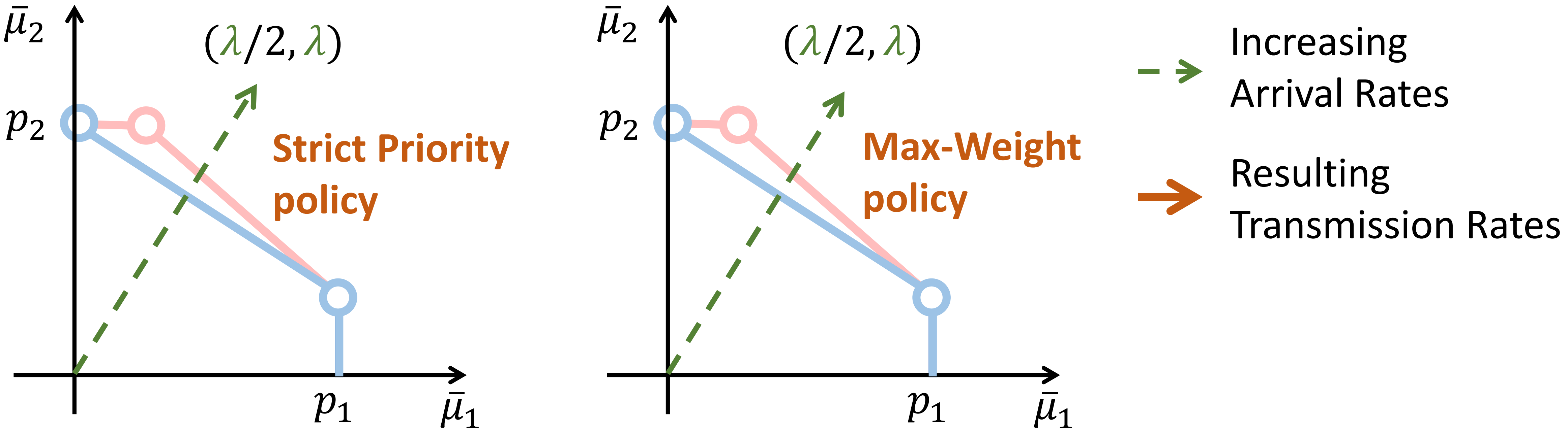
- Max-Weight policy: transmits queue with $s_n(t) = ON$ and largest backlog $Q_n(t)$.
- Max-Weight is throughput-optimal. [to be proven in this lecture] Balances between exploring good channel conditions and serving the largest queue.

Outline

- Multi-commodity flow problem
- Recap from previous lectures
- **Overloaded system**
  - Problem Statement
  - Utility Function
  - Drift-Plus Penalty Algorithm
    - Admission Control
    - Routing Policy
    - Scheduling Policy
  - Performance Analysis and Optimality Results

Overloaded system – Example

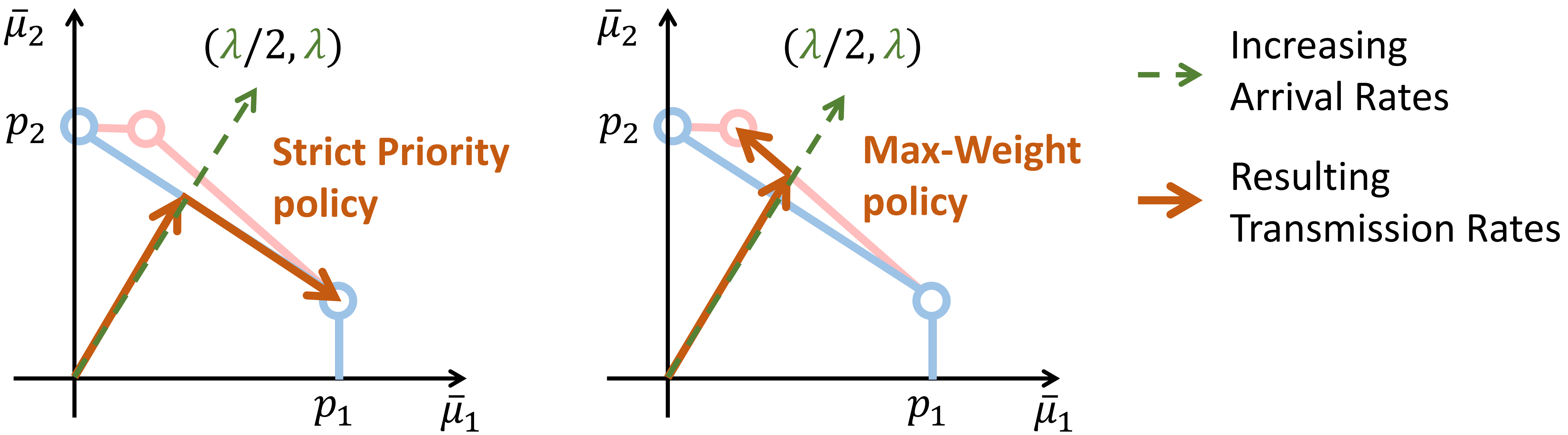
- Let $p_1 = 0.5,\ p_2 = 0.6$ and $(\lambda_1, \lambda_2) = (\lambda/2, \lambda)$ for increasing values of $\lambda \geq 0$.
- Outside of the capacity region Λ, both $Q_1(t)$ and $Q_2(t)$ are unstable.



- Strict Priority policy always serves $Q_1(t)$ first. What happens when $\lambda \to \infty$.
- How does the Max-Weight policy behaves?
- Which one is “better”? What would be the desirable outcome?

Overloaded system – Example

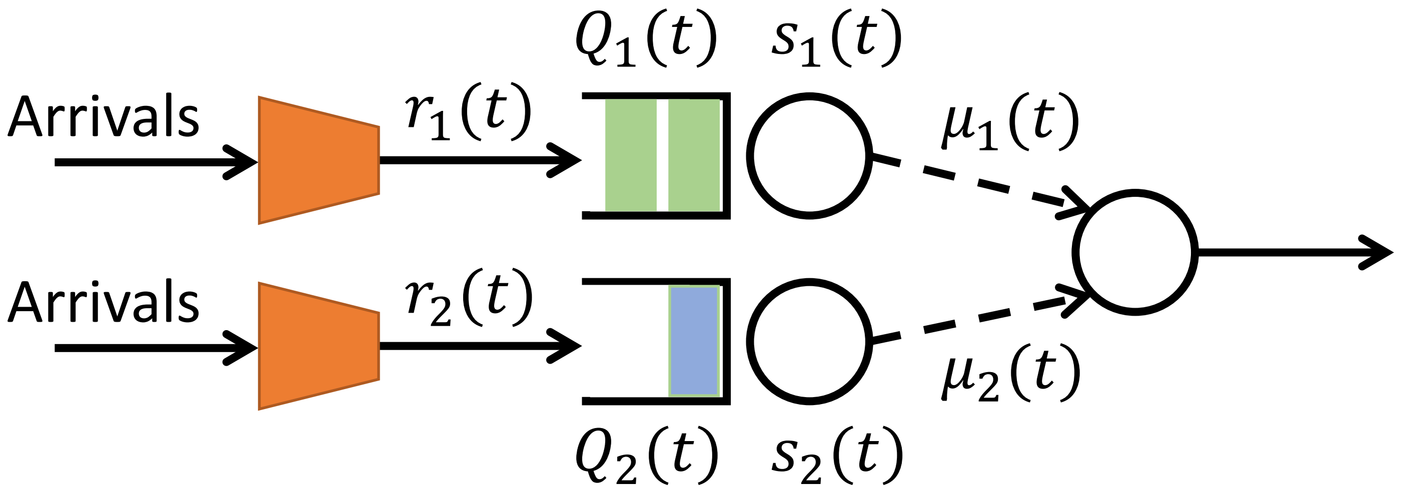
- Let $p_1 = 0.5,\ p_2 = 0.6$ and $(\lambda_1, \lambda_2) = (\lambda/2, \lambda)$ for increasing values of $\lambda \geq 0$.
- Outside of the capacity region Λ, both $Q_1(t)$ and $Q_2(t)$ are unstable.



- <u>Strict Priority policy</u> always serves $Q_1(t)$ first. That is why it goes to the RHS when $\lambda \to \infty$.
- <u>Max-Weight policy</u> serves $Q_2(t)$ first because $Q_2(t) > Q_1(t)$ as $t \to \infty$ and $\lambda \to \infty$.
- Which one is "better"? What would be the desirable outcome?

Overloaded system – Definition

- Assume that arrival rates λ_n^c are **infinitely large**. Then all queues $Q_n^c(t)$ are unstable.
- **Admission Control.** Let $r_n^c(t)$ be the number of packets admitted to $Q_n^c(t)$ at time t.
  - Assume that admission is bounded $\sum_{c=1}^{K} r_n^c(t) \leq R_n^{max}, \forall t, n$;
  - Define the time-average admission rate as $\bar{r}_n^c := \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[r_n^c(\tau)\right], \forall n, c$;



- <u>From another perspective</u>: now we can control packet arrivals to the queues $r_n^c(t)$. Can we utilize admission control to achieve a "desired network behavior"?

Overloaded system – Utility Function

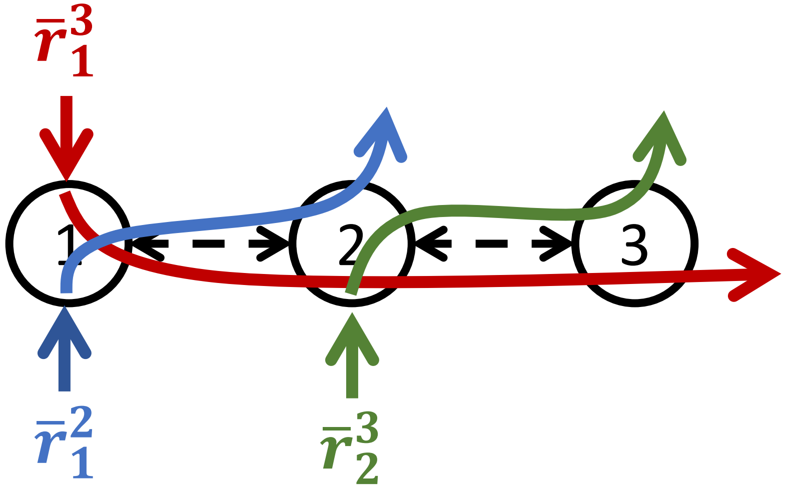
- Let $g_n^c(r)$ be **strictly concave, non-decreasing and continuously differentiable.**
  - Capture satisfaction/utility attained from sending commodity-c at a time-average rate r.
  - Can be used to achieve fairness across commodities and nodes.
  - Good model for elastic flows (e.g. file download). Not good for inelastic flow or for flows with an intrinsic rate (e.g. real-time video).

- Example: consider a network with 3 nodes, 3 flows and equal link capacities of 1. What is the transmission rate distribution that maximizes:

  1) $\bar{r}_1^2 + \bar{r}_1^3 + \bar{r}_2^3$? $(\bar{r}_1^2; \bar{r}_1^3; \bar{r}_2^3) = (1; 0; 1)$

  2) $log(\bar{r}_1^2) + log(\bar{r}_1^3) + log(\bar{r}_2^3)$? *Ans.* $\left(\frac{2}{3}; \frac{1}{3}; \frac{2}{3}\right)$

  3) *Max-min fairness ? Ans.* $\left(\frac{1}{2}; \frac{1}{2}; \frac{1}{2}\right)$



Overloaded system – Goal

- Recall that $\mu_{n,j}^{c}(t)$ is the **offered** transmission opportunity over link (n, j) to commodity-c. Notice that: $\sum_{c=1}^{K} \mu_{n,j}^{c}(t) = \mu_{n,j}(t)$ is the total **offered** transmission opportunity.

- With controlled packet arrivals to the queues $r_n^c(t)$, we have

$$Q_n^c(t+1) \leq \max\left\{Q_n^c(t) - \sum_{j=1}^{N} \mu_{n,j}^{c}(t)\,;0\right\} + \sum_{i=1}^{N} \mu_{i,n}^{c}(t) + r_n^c(t)$$

- **<u>Goal</u>** is to design admission, routing and scheduling algorithms that solve the optimal sum utility problem:

$$\max \sum_{n=1}^{N} \sum_{c=1}^{K} g_n^c(\bar{r}_n^c) \quad s.t.: (\bar{r}_n^c)_{n,c} \in \Lambda \;\text{ and }\; \bar{r}_n^c \geq 0, \forall n, c$$

- Is it possible to use Stationary Randomized Policy? Is it a practical policy?

Lyapunov Optimization

- Lindley recursion: $Q_n^c(t+1) \leq \max\left\{Q_n^c(t) - \sum_{j=1}^{N} \mu_{n,j}^c(t)\,; 0\right\} + \sum_{i=1}^{N} \mu_{i,n}^c(t) + r_n^c(t)$
- Lyapunov Function: $L(t) = \frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{K}\left(Q_n^c(t)\right)^2$
- One-slot Lyapunov Drift: $\Delta\big(Q(t)\big) = \mathbb{E}[L(t+1) - L(t)|Q(t)]$
- Drift-Plus Penalty (DPP) Function: $\Delta\big(Q(t)\big) + V\,\mathbb{E}\left[-\sum_{n=1}^{N}\sum_{c=1}^{K} g_n^c\big(r_n^c(t)\big)\,\middle|\,Q(t)\right]$
- Next, we obtain an **upper bound** to the DPP Function and derive an **algorithm that minimizes** this upper bound. By minimizing the upper bound, we aim to achieve low sum of queues backlogs $Q_n^c(t)$ and high sum of utility functions $g_n^c(\bar{r}_n^c)$. **The DPP algorithm is throughput optimal and ensures that utility is arbitrarily close to optimal.**

- Manipulating Lindley Recursion

$$\left(Q_n^c(t+1)\right)^2 \leq \boldsymbol{max}\left\{\boldsymbol{Q_n^c(t)} - \sum_{j=1}^{N} \boldsymbol{\mu_{n,j}^c(t)}\,;\boldsymbol{0}\right\}^{\boldsymbol{2}} + \left(\sum_{i=1}^{N} \mu_{i,n}^c(t) + r_n^c(t)\right)^2 +$$

$$+2\left(\sum_{i=1}^{N} \mu_{i,n}^c(t) + r_n^c(t)\right)\boldsymbol{max}\left\{\boldsymbol{Q_n^c(t)} - \sum_{j=1}^{N} \boldsymbol{\mu_{n,j}^c(t)}\,;\boldsymbol{0}\right\}$$

$$\left(Q_n^c(t+1)\right)^2 - \left(\boldsymbol{Q_n^c(t)}\right)^{\boldsymbol{2}} \leq -\boldsymbol{2Q_n^c(t)}\sum_{j=1}^{N} \boldsymbol{\mu_{n,j}^c(t)} + \left(\sum_{j=1}^{N} \boldsymbol{\mu_{n,j}^c(t)}\right)^{\boldsymbol{2}} +$$

$$+\left(\sum_{i=1}^{N} \mu_{i,n}^c(t) + r_n^c(t)\right)^2 + 2\left(\sum_{i=1}^{N} \mu_{i,n}^c(t) + r_n^c(t)\right)\boldsymbol{Q_n^c(t)}$$

$$\left(Q_n^c(t+1)\right)^2 - \left(Q_n^c(t)\right)^2 \leq -2Q_n^c(t)\left[\sum_{j=1}^{N} \boldsymbol{\mu}_{n,j}^{c}(\boldsymbol{t}) - \sum_{i=1}^{N} \boldsymbol{\mu}_{i,n}^{c}(\boldsymbol{t}) - \boldsymbol{r}_{n}^{c}(\boldsymbol{t})\right] + \\ + \left(\sum_{j=1}^{N} \boldsymbol{\mu}_{n,j}^{c}(\boldsymbol{t})\right)^2 + \left(\sum_{i=1}^{N} \boldsymbol{\mu}_{i,n}^{c}(\boldsymbol{t}) + \boldsymbol{r}_{n}^{c}(\boldsymbol{t})\right)^2$$

- Substituting into the Lyapunov Drift: $\Delta\left(Q(t)\right) = \frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[\left(Q_n^c(t+1)\right)^2 - \left(Q_n^c(t)\right)^2 \middle| Q(t)\right]$

$$\Delta(t) \leq -\sum_{n=1}^{N}\sum_{c=1}^{K} Q_n^c(t)\mathbb{E}\left[\sum_{j=1}^{N} \boldsymbol{\mu}_{n,j}^{c}(\boldsymbol{t}) - \sum_{i=1}^{N} \boldsymbol{\mu}_{i,n}^{c}(\boldsymbol{t}) - \boldsymbol{r}_{n}^{c}(\boldsymbol{t}) \middle| Q(t)\right] + B$$

where $B \geq \frac{1}{2}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[\left(\sum_{j=1}^{N} \boldsymbol{\mu}_{n,j}^{c}(\boldsymbol{t})\right)^2 + \left(\sum_{i=1}^{N} \boldsymbol{\mu}_{i,n}^{c}(\boldsymbol{t}) + \boldsymbol{r}_{n}^{c}(\boldsymbol{t})\right)^2 \middle| Q(t)\right]$

- Assuming that second moments are all bounded, B is a constant.

- Drift-Plus Penalty: consider the expression $\Delta(t) - V\,\mathbb{E}\left[\sum_{n=1}^{N}\sum_{c=1}^{K} g_n^c\left(\boldsymbol{r_n^c(t)}\right) \middle| Q(t)\right]$

$$\Delta(t) - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right] \leq B - \sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[Q_n^c(t)\sum_{j=1}^{N}\boldsymbol{\mu_{n,j}^c(t)} - Q_n^c(t)\sum_{i=1}^{N}\boldsymbol{\mu_{i,n}^c(t)} + \right.$$

$$\left. -Q_n^c(t)\boldsymbol{r_n^c(t)} + V g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right]$$

$$\Delta(t) - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right] \leq B - \sum_{n=1}^{N}\sum_{c=1}^{K}Q_n^c(t)\mathbb{E}\left[\sum_{j=1}^{N}\boldsymbol{\mu_{n,j}^c(t)} - \sum_{i=1}^{N}\boldsymbol{\mu_{i,n}^c(t)}\middle|Q(t)\right] +$$

$$-\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[-Q_n^c(t)\boldsymbol{r_n^c(t)} + V g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right]$$

- **DPP algorithm minimizes the upper bound on the RHS at every slot *t*.** The minimization can be separated into **two sub-problems**: i) routing and scheduling $\boldsymbol{\mu_{i,j}^C(t)}$; and ii) admission control $\boldsymbol{r_n^c(t)}$.

Drift-Plus Penalty

The DPP minimization can be separated into two sub-problems:

- Routing & Scheduling: $$\max_{\underline{\mu}(t)\in\Gamma}\left\{\sum_{n=1}^{N}\sum_{c=1}^{K} Q_n^c(t)\mathbb{E}\left[\sum_{j=1}^{N}\boldsymbol{\mu_{n,j}^c(t)}-\sum_{i=1}^{N}\boldsymbol{\mu_{i,n}^c(t)}\,\middle|Q(t)\right]\right\}$$

- Admission Control: $$\max_{r}\left\{\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[-Q_n^c(t)\boldsymbol{r_n^c(t)}+V g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right]\right\}$$
$$s.t.: \sum_{c=1}^{K}\boldsymbol{r_n^c(t)}\leq R_n^{max},\forall n$$
$$\boldsymbol{r_n^c(t)}\geq 0,\forall n,c$$

Drift-Plus Penalty – Routing & Scheduling

- Routing & Scheduling

$$\max_{\underline{\mu}(t)\in\Gamma}\left\{\sum_{n=1}^{N}\sum_{c=1}^{K} Q_n^c(t)\mathbb{E}\left[\sum_{j=1}^{N}\boldsymbol{\mu_{n,j}^c(t)} - \sum_{i=1}^{N}\boldsymbol{\mu_{i,n}^c(t)}\,\middle|\,Q(t)\right]\right\}$$

$$\max_{\underline{\mu}(t)\in\Gamma}\left\{\sum_{n=1}^{N}\sum_{c=1}^{K}\sum_{j=1}^{N}\mathbb{E}\left[Q_n^c(t)\boldsymbol{\mu_{n,j}^c(t)}\middle|Q(t)\right] - \sum_{n=1}^{N}\sum_{c=1}^{K}\sum_{i=1}^{N}\mathbb{E}\left[Q_n^c(t)\boldsymbol{\mu_{i,n}^c(t)}\middle|Q(t)\right]\right\}$$

$$\max_{\underline{\mu}(t)\in\Gamma}\left\{\sum_{i=1}^{N}\sum_{c=1}^{K}\sum_{j=1}^{N}\mathbb{E}\left[Q_i^c(t)\boldsymbol{\mu_{i,j}^c(t)}\middle|Q(t)\right] - \sum_{j=1}^{N}\sum_{c=1}^{K}\sum_{i=1}^{N}\mathbb{E}\left[Q_j^c(t)\boldsymbol{\mu_{i,j}^c(t)}\middle|Q(t)\right]\right\}$$

$$\max_{\underline{\mu}(t)\in\Gamma}\left\{\sum_{i=1}^{N}\sum_{c=1}^{K}\sum_{j=1}^{N}\left(Q_i^c(t) - Q_j^c(t)\right)\mathbb{E}\left[\boldsymbol{\mu_{i,j}^c(t)}\middle|Q(t)\right]\right\}$$

Drift-Plus Penalty – Routing & Scheduling

Routing & Scheduling
$$\max_{\underline{\mu}(t)\in\Gamma}\left\{\sum_{i=1}^{N}\sum_{c=1}^{K}\sum_{j=1}^{N}\left(Q_i^c(t)-Q_j^c(t)\right)\mathbb{E}\left[\boldsymbol{\mu_{i,j}^c(t)}\middle|Q(t)\right]\right\}$$

Solution [**Backpressure** – presented in previous lectures]:

- Routing: at time t and for every link (i,j), select the commodity with highest differential backlog, namely $c_{i,j}^* = \text{argmax}\{Q_i^c(t) - Q_j^c(t)\}$
- Scheduling: for a given state $s(t) = s$, select action $I(t) = I$ such that the set of transmission rates $\boldsymbol{U}(s, I) = [\boldsymbol{\mu_{i,j}(t)}]_{i,j}$ yields maximum sum:
$$\sum_{(i,j)}\left(Q_i^{c_{i,j}^*}(t) - Q_j^{c_{i,j}^*}(t)\right)\boldsymbol{\mu_{i,j}(t)}$$

Notice that full rate is allocated to commodity $c_{i,j}^*$, namely $\boldsymbol{\mu_{i,j}(t)} = \boldsymbol{\mu_{i,j}^{c_{i,j}^*}(t)}$.

Drift-Plus Penalty – Admission Control

- Admission Control

$$\max_{r}\left\{\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[-Q_n^c(t)\boldsymbol{r_n^c(t)}+Vg_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right]\right\}\quad s.t.:\sum_{c=1}^{K}\boldsymbol{r_n^c(t)}\le R_n^{max},\forall n$$

$$\boldsymbol{r_n^c(t)}\ge 0,\forall n,c$$

- Maximization is **separable into a per-node problem**. At time t, each node n should select the set of values $\boldsymbol{r_n^c(t)}, \forall c$, that solve the problem:

$$\max_{r_n^c(t)}\left\{\sum_{c=1}^{K}\left[Vg_n^c\left(\boldsymbol{r_n^c(t)}\right)-Q_n^c(t)\boldsymbol{r_n^c(t)}\right]\right\}\quad s.t.:\sum_{c=1}^{K}\boldsymbol{r_n^c(t)}\le R_n^{max}$$

$$\boldsymbol{r_n^c(t)}\ge 0,\forall n,c$$

Each node solves the problem independently of other nodes. The objective function is concave and constraints are linear. **How to solve?**

Drift-Plus Penalty – KKT Conditions

- Lagrangean:

$$\mathcal{L}(\boldsymbol{r_n^c(t)}, \eta, \gamma^c) = \sum_{c=1}^{K}\left[V g_n^c\left(\boldsymbol{r_n^c(t)}\right) - Q_n^c(t)\boldsymbol{r_n^c(t)}\right] - \eta\left(\sum_{c=1}^{K}\boldsymbol{r_n^c(t)} - R_n^{max}\right) + \sum_{c=1}^{K}\gamma^c\boldsymbol{r_n^c(t)}$$

where η and γ^c are non-negative KKT multipliers. The KKT Conditions can be written as:

(Stationarity) $\nabla_{\boldsymbol{r_n^c(t)}}\mathcal{L}(.) = V\left(g_n^c\left(\boldsymbol{r_n^c(t)}\right)\right)' - Q_n^c(t) - \eta + \gamma^c = 0$

(Complementary Slackness) $\eta\left(\sum_{c=1}^{K}\boldsymbol{r_n^c(t)} - R_n^{max}\right) = 0$ and $\gamma^c\boldsymbol{r_n^c(t)} = 0, \forall c$

(Primal/Dual Feasibility) $\sum_{c=1}^{K}\boldsymbol{r_n^c(t)} \leq R_n^{max}$ and $\boldsymbol{r_n^c(t)} \geq 0, \forall c$ and $\eta \geq 0$ and $\gamma^c \geq 0, \forall c$

Drift-Plus Penalty – Solution

- From Stationarity: $\nabla_{\boldsymbol{r_n^c(t)}}\mathcal{L}(.) = 0 \;\Rightarrow\; \left(g_n^c\left(\boldsymbol{r_n^c(t)}\right)\right)' = \frac{Q_n^c(t) + \eta - \gamma^c}{V}$ (Eq.1)
- From Complementary Slackness, if $\gamma^c \neq 0$, then $\boldsymbol{r_n^c(t)} = 0$.
- Initially, assume that $\gamma^c = 0, \forall c$. Then we know that $\left(g_n^c\left(\boldsymbol{r_n^c(t)}\right)\right)' = \frac{Q_n^c(t) + \eta}{V}, \forall c$
- Notice that $\left(g_n^c(r)\right)'$ is non-increasing, invertible and **that ↑ $\boldsymbol{\eta}$ always leads to ↓ $\boldsymbol{r}$**.
- Algorithm:
  1) Initialization: $\eta = 0$
  2) Find $\left(\boldsymbol{r_n^c(t)}\right)_{c=1}^{K}$ associated with η using (Eq.1). If $\boldsymbol{r_n^c(t)} < 0$, then set $\boldsymbol{r_n^c(t)} = 0$.
  3) If $\sum_{c=1}^{K} \boldsymbol{r_n^c(t)} \leq R_n^{max}$ then the **unique** solution $\left(\boldsymbol{r_n^c(t)}\right)_{c=1}^{K}$ was found.
  4) Otherwise, increase η slightly and go back to step 2.

Drift-Plus Penalty – Solution (Example)

Example: consider the utility function $g_n^c(\boldsymbol{r_n^c(t)}) = \log(\boldsymbol{r_n^c(t)})$.

- Then $\left(g_n^c(\boldsymbol{r_n^c(t)})\right)' = \dfrac{1}{\boldsymbol{r_n^c(t)}} = \dfrac{Q_n^c(t) + \eta - \gamma^c}{V} \rightarrow \boldsymbol{r_n^c(t)} = \dfrac{V}{Q_n^c(t) + \eta - \gamma^c}, \forall n, c, t$

- *Assuming* $\eta = \gamma^c = 0$ *and that* R_n^{max} *is large enough, the solution is:*

$$\boldsymbol{r_n^c(t)} = \frac{V}{Q_n^c(t)}, \forall n, c, t$$

- Notice that:
  - A larger backlog on the queue ↑ $\mathrm{Q_n^c(t)}$ leads to less packets being admitted to the queue ↓ $\mathbf{r_n^c(t)}$, what makes sense.
  - Larger V implies in larger $\mathbf{r_n^c(t)}$ which, in turn, implies in more network congestion.

Drift-Plus Penalty Algorithm

At every time t, the DPP algorithm runs three steps:

- Routing: for every link (i, j), select the commodity with highest differential backlog, namely

$$c_{i,j}^{*} = \operatorname{argmax}\{Q_i^c(t) - Q_j^c(t)\}$$

- Scheduling: for a given state $s(t) = s$, select action $I(t) = I$ such that the set of transmission rates $\boldsymbol{U}(s, I) = [\boldsymbol{\mu_{i,j}(t)}]_{i,j}$ yields maximum sum:

$$\sum_{(i,j)} \left(Q_i^{c_{i,j}^{*}}(t) - Q_j^{c_{i,j}^{*}}(t) \right) \boldsymbol{\mu_{i,j}(t)}$$

- Admission control: each node n uses its own KKT Conditions to attain the values of $\boldsymbol{r_n^c(t)}$ which solve:

$$\max_{\boldsymbol{r_n^c(t)}} \left\{ \sum_{c=1}^{K} \left[V g_n^c\left(\boldsymbol{r_n^c(t)}\right) - Q_n^c(t) \boldsymbol{r_n^c(t)} \right] \right\} \quad s.t.: \sum_{c=1}^{K} \boldsymbol{r_n^c(t)} \leq R_n^{max} ; \; \boldsymbol{r_n^c(t)} \geq 0, \forall n, c$$

Drift-Plus Penalty Algorithm

- **Admission Control**. The parameter V captures the emphasis on utility maximization. If V is large, then the admitted rates tend to be large, thus increasing the utility, but at the same time increasing the network delay caused by congestion. Notice that admission control on node n only requires information available locally.

- **Routing & Scheduling** is done using the Backpressure policy (already discussed in previous lectures). Recall that routing requires no pre-specified paths, since paths are learned dynamically. Moreover, this algorithm does not need information about arrival rates or channel state statistics.

- Next, we show that the **DPP Algorithm is throughput optimal and ensures that utility is arbitrarily close to optimal.**

Performance Analysis – Randomized Policy

- Prior to analyzing the performance of the DPP algorithm, we assess the **Stationary Randomized Policy** associated with our original utility maximization problem:

$$\max \sum_{n=1}^{N} \sum_{c=1}^{K} g_n^c(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}}) \quad s.t.: (\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}})_{n,c} \in \Lambda \text{ and } \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}} \geq 0, \forall n, c$$

- Denote $(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*})_{n,c}$ as the optimal solution to the utility maximization. A simple admission control algorithm that achieves the optimal solution is: $\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*} = r_n^c(t), \forall t.$

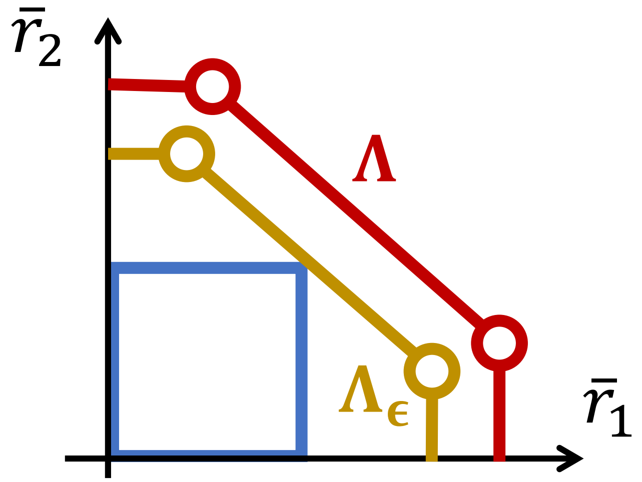
- We know from our discussion of the capacity region that for $(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*})_{n,c} \in \Lambda,$ there exists flow variables $\bar{f}_{i,j}^c$ such that $\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*} + \sum_{i=1}^{N} \bar{f}_{i,n}^c = \sum_{j=1}^{N} \bar{f}_{n,j}^c$ and a **Stationary Randomized Policy** with time-average packet transmission rates $\bar{\boldsymbol{\mu}}_{\boldsymbol{i,j}}^{\boldsymbol{c}}$ such that $\bar{f}_{i,j}^c \leq \bar{\boldsymbol{\mu}}_{\boldsymbol{i,j}}^{\boldsymbol{c}}, \forall i, j, c.$

- Consider the **Stationary Randomized Policy with rates** $\bar{\boldsymbol{\mu}}_{\boldsymbol{i,j}}^{\boldsymbol{c}} = \bar{f}_{i,j}^c, \forall i, j, c.$

Performance Analysis – Near-optimal solution

- **Near-optimal solution.** Let $\epsilon > 0$ and define the set $\Lambda_\epsilon = \{\bar{r}_n^c \mid (\bar{r}_n^c + \epsilon) \in \Lambda\}$.
- Consider the near-optimal solution $\left(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon})\right)_{n,c}$ to the utility maximization problem:

$$\max \sum_{n=1}^{N} \sum_{c=1}^{K} g_n^c(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}}) \quad s.t.: (\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}})_{n,c} \in \Lambda_\epsilon \text{ and } \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}} \geq 0, \forall n, c$$



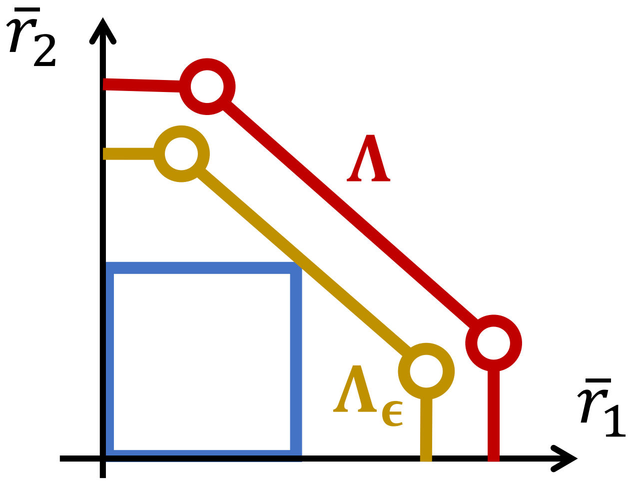
- **Lemma:** if $g_n^c(r)$ are non-negative and concave, and if there is a scalar $r_{min} > 0$ such that a hypercube with edge size r_{min} can fit into Λ, then:

$$\sum_{n=1}^{N} \sum_{c=1}^{K} g_n^c\left(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon})\right) \to \sum_{n=1}^{N} \sum_{c=1}^{K} g_n^c(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}) \quad \text{as} \quad \epsilon \to 0$$

Performance Analysis – Randomized Policy

- **Near-optimal solution.** Let $\epsilon > 0$ and define the set $\Lambda_\epsilon = \{\bar{r}_n^c \mid (\bar{r}_n^c + \epsilon) \in \Lambda\}$.
- Consider the near-optimal solution $\left(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon})\right)_{n,c}$ to the utility maximization problem:

$$\max \sum_{n=1}^{N} \sum_{c=1}^{K} g_n^c(\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}}) \quad s.t.: (\bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}})_{n,c} \in \Lambda_\epsilon \text{ and } \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}} \geq 0, \forall n, c$$



- Consider the **Stationary Randomized Policy:** $\bar{r}_n^c(t) = \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon}), \forall n, c, t$ and $\bar{\boldsymbol{\mu}}_{\boldsymbol{i,j}}^{\boldsymbol{c}} = \bar{f}_{i,j}^c, \forall i, j, c$. Recall that $\sum_{i=1}^N \bar{f}_{i,n}^c + \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*} = \sum_{j=1}^N \bar{f}_{n,j}^c$. Hence, it follows that:

$$\sum_{i=1}^{N} \bar{\boldsymbol{\mu}}_{\boldsymbol{i,n}}^{\boldsymbol{c}} + \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon}) + \epsilon \leq \sum_{j=1}^{N} \bar{\boldsymbol{\mu}}_{\boldsymbol{n,j}}^{\boldsymbol{c}} \Rightarrow \sum_{j=1}^{N} \bar{\boldsymbol{\mu}}_{\boldsymbol{n,j}}^{\boldsymbol{c}} - \sum_{i=1}^{N} \bar{\boldsymbol{\mu}}_{\boldsymbol{i,n}}^{\boldsymbol{c}} - \bar{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon}) \geq \epsilon > 0 \quad \text{(Eq2)}$$

- Next, we compare the drift of the DPP algorithm with the drift of the Randomized Policy.

Performance Analysis – DPP Algorithm

- Recall the expression for the Drift-Plus Penalty:

$$\Delta(t) - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right] \leq B - \sum_{n=1}^{N}\sum_{c=1}^{K} Q_n^c(t)\mathbb{E}\left[\sum_{j=1}^{N}\boldsymbol{\mu_{n,j}^c(t)} - \sum_{i=1}^{N}\boldsymbol{\mu_{i,n}^c(t)}\,\middle|Q(t)\right] + -\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[-Q_n^c(t)\boldsymbol{r_n^c(t)} + V g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right]$$

- By definition, this **upper bound is minimizes by the DPP Algorithm**. Hence, the Stationary Randomized Policy achieves a larger upper bound. Substituting $\boldsymbol{\bar{\mu}_{i,j}^c}$ and $\boldsymbol{\bar{r}_n^{c*}(\epsilon)}$:

$$\Delta(t) - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right] \leq B - \sum_{n=1}^{N}\sum_{c=1}^{K} Q_n^c(t)\left[\sum_{j=1}^{N}\boldsymbol{\bar{\mu}_{n,j}^c} - \sum_{i=1}^{N}\boldsymbol{\bar{\mu}_{n,j}^c}\right] + -\sum_{n=1}^{N}\sum_{c=1}^{K}\left[-Q_n^c(t)\boldsymbol{\bar{r}_n^{c*}(\epsilon)} + V g_n^c\left(\boldsymbol{\bar{r}_n^{c*}(\epsilon)}\right)\right]$$

- By rearranging the upper bound and utilizing (Eq.2), we have:

$$\Delta(t) - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\middle|Q(t)\right] \leq B - \epsilon\sum_{n=1}^{N}\sum_{c=1}^{K}Q_n^c(t) - \sum_{n=1}^{N}\sum_{c=1}^{K}\left[V g_n^c\left(\boldsymbol{\bar{r}_n^{c*}(\epsilon)}\right)\right]$$

- Taking the expectation w.r.t $Q_n^c(t)$ and using the definition of Lyapunov Drift:

$$\mathbb{E}[L(t+1)] - \mathbb{E}[L(t)] - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\right] \leq B - \epsilon\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}[Q_n^c(t)] - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{\bar{r}_n^{c*}(\epsilon)}\right)\right]$$

- Summing over $t \in \{0,2,\dots,T-1\}$ and dividing by T gives:

$$\frac{\mathbb{E}[L(T)]}{T} - \frac{\mathbb{E}[L(0)]}{T} - \frac{V}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r_n^c(t)}\right)\right] \leq$$

$$\leq B - \frac{\epsilon}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}[Q_n^c(t)] - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{\bar{r}_n^{c*}(\epsilon)}\right)\right]$$

- Rearranging the expression and knowing that $\mathbb{E}[L(T)]/T$ is non-negative:

$$\frac{\epsilon}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}[Q_n^c(t)] - \frac{V}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\boldsymbol{r}_{\boldsymbol{n}}^{\boldsymbol{c}}(\boldsymbol{t})\right)\right] \le B - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\overline{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon})\right)\right] + \frac{\mathbb{E}[L(0)]}{T}$$

- Taking the limit $T \to \infty$ and assuming that $\mathbb{E}[L(0)]$ is finite gives:

$$\sum_{n=1}^{N}\sum_{c=1}^{K}\lim_{T\to\infty}\left[\frac{\epsilon}{T}\sum_{t=0}^{T-1}\mathbb{E}[Q_n^c(t)]\right] - \sum_{n=1}^{N}\sum_{c=1}^{K}\lim_{T\to\infty}\left[\frac{V}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[g_n^c\left(\boldsymbol{r}_{\boldsymbol{n}}^{\boldsymbol{c}}(\boldsymbol{t})\right)\right]\right] \le B - V\sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\overline{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon})\right)\right]$$

- **Conclusion 1:**

$$\sum_{n=1}^{N}\sum_{c=1}^{K}\lim_{T\to\infty}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[Q_n^c(t)]\right] \le \frac{B}{\epsilon} + \frac{V}{\epsilon}\sum_{n=1}^{N}\sum_{c=1}^{K}\left\{\lim_{T\to\infty}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[g_n^c\left(\boldsymbol{r}_{\boldsymbol{n}}^{\boldsymbol{c}}(\boldsymbol{t})\right)\right]\right] - \mathbb{E}\left[g_n^c\left(\overline{\boldsymbol{r}}_{\boldsymbol{n}}^{\boldsymbol{c}*}(\boldsymbol{\epsilon})\right)\right]\right\} < \infty$$

- **All queues $\boldsymbol{Q_n^c(t)}$ are strongly stable. DPP algorithm is throughput optimal.**

- On the other hand: $$\sum_{n=1}^{N}\sum_{c=1}^{K}\lim_{T\to\infty}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[g_n^c\left(\boldsymbol{r}_n^c(\boldsymbol{t})\right)\right]\right] \geq \sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\bar{\boldsymbol{r}}_n^{c*}(\boldsymbol{\epsilon})\right)\right] - \frac{B}{V}$$

- By Jensen's inequality and by concavity and continuity of $g_n^c\left(\boldsymbol{r}_n^c(\boldsymbol{t})\right)$, we have:

$$\sum_{n=1}^{N}\sum_{c=1}^{K}\lim_{T\to\infty} g_n^c\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\boldsymbol{r}_n^c(\boldsymbol{t})]\right) \geq \sum_{n=1}^{N}\sum_{c=1}^{K}\lim_{T\to\infty}\left[\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[g_n^c\left(\boldsymbol{r}_n^c(\boldsymbol{t})\right)\right]\right] \geq \sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\bar{\boldsymbol{r}}_n^{c*}(\boldsymbol{\epsilon})\right)\right] - \frac{B}{V}$$

$$\sum_{n=1}^{N}\sum_{c=1}^{K} g_n^c\left(\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\boldsymbol{r}_n^c(\boldsymbol{t})]\right) \geq \sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c\left(\bar{\boldsymbol{r}}_n^{c*}(\boldsymbol{\epsilon})\right)\right] - \frac{B}{V}$$

- **Conclusion 2:** For $\epsilon \to 0$: $$\sum_{n=1}^{N}\sum_{c=1}^{K} g_n^c(\bar{\boldsymbol{r}}_n^c) \geq \sum_{n=1}^{N}\sum_{c=1}^{K}\mathbb{E}\left[g_n^c(\bar{\boldsymbol{r}}_n^{c*})\right] - \frac{B}{V}$$

- **Larger values of V take the DPP Algorithm arbitrarily close to the optimal utility.**

Topics covered

- Definition of the Multi-commodity flow problem and discussion about Queue Stability, Capacity Region and Stationary Randomized policies.
- Discussion about Utility Function and Fairness.
- Development of the Drift-Plus Penalty algorithm for an overloaded system. In particular, we described an Admission Control Algorithm, a Routing Policy and a Scheduling Policy,
- Performance analysis of the Drift-Plus Penalty algorithm. Under mild assumptions, it was shown to stabilize all queues in the network and at the same time achieve utility which is arbitrarily close to the optimal.

References

Slides adapted from Dr. Chih-Ping Li's lecture.

- M. J. Neely, "Dynamic power allocation and routing for satellite and wireless networks with time varying channels," PhD thesis, MIT 2004
- L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," IEEE Trans. Automatic Control, 37(12):1936-1948, Dec. 1992
- L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," IEEE Trans. Information Theory, 39(2):466-478, March 1993
- Michael J. Neely, Eytan Modiano, and Chih-ping Li, "Fairness and optimal stochastic control for heterogeneous networks," IEEE Trans. Networking, 16(2):396-409, April 2008
- L. Georgiadis, M. J. Neely, L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," NOW Foundations and Trends in Networking, 2006
- M. J. Neely, "Stochastic network optimization with applications to communication and queueing systems," Morgan&Claypool, 2010