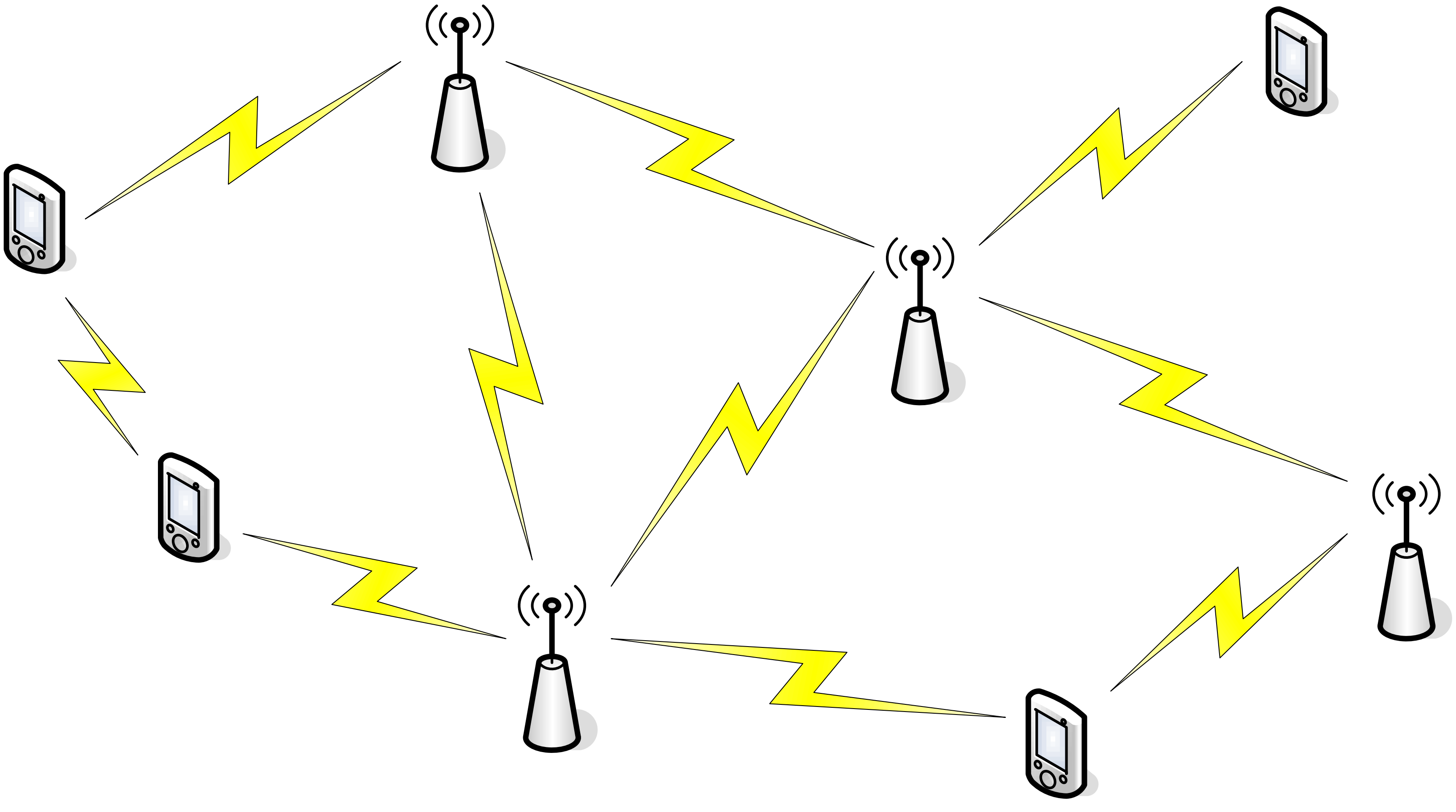
Optimization, Queueing and Resource Allocation in Wireless Networks

R. Srikant
Department of ECE & CSL
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Collaborators: Atilla Eryilmaz (OSU), Juan Jose Jaramillo (Illinois), Shihuan Liu and Lei Ying (Iowa State)

Wireless network



High-Level Goal

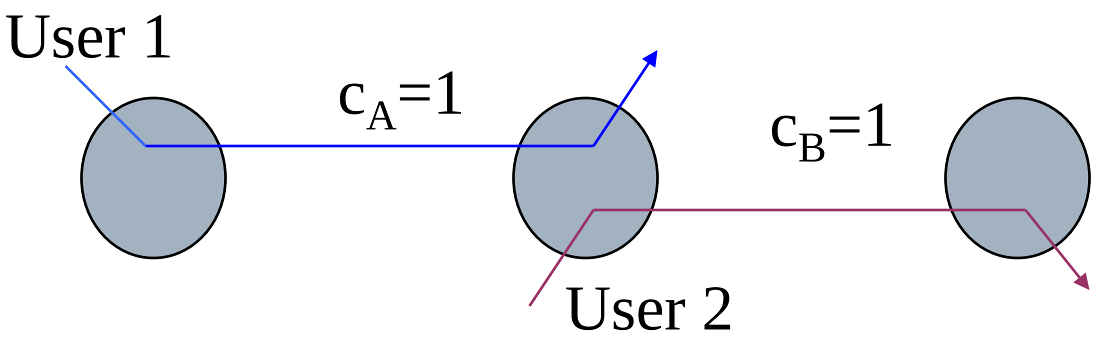
- Different types of traffic sharing the wireless network:
  - Unicast and multicast
  - Short-lived flows and long-lived flows
  - Elastic and Inelastic
  - Non-real-time and Real-time (with delay & jitter requirements)

- Need an *efficient protocol stack* to allocate resources between these different types of flows.

Outline of the Talk

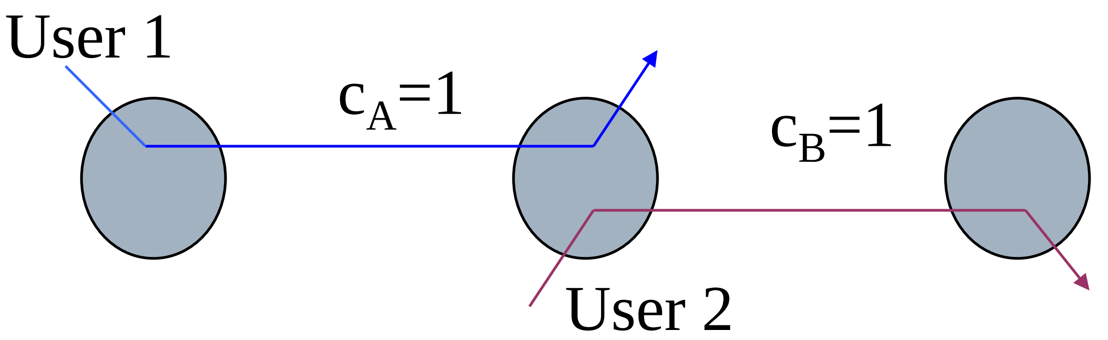
- Basic Theory (2005)
  - Optimization and Resource Allocation
  - Traditional results for long-lived elastic flows only
- New Results (2009)
  - Packets with strict deadlines
  - Mixture of flows with finite sizes and persistent flows

2-Link, 2-User wireless network



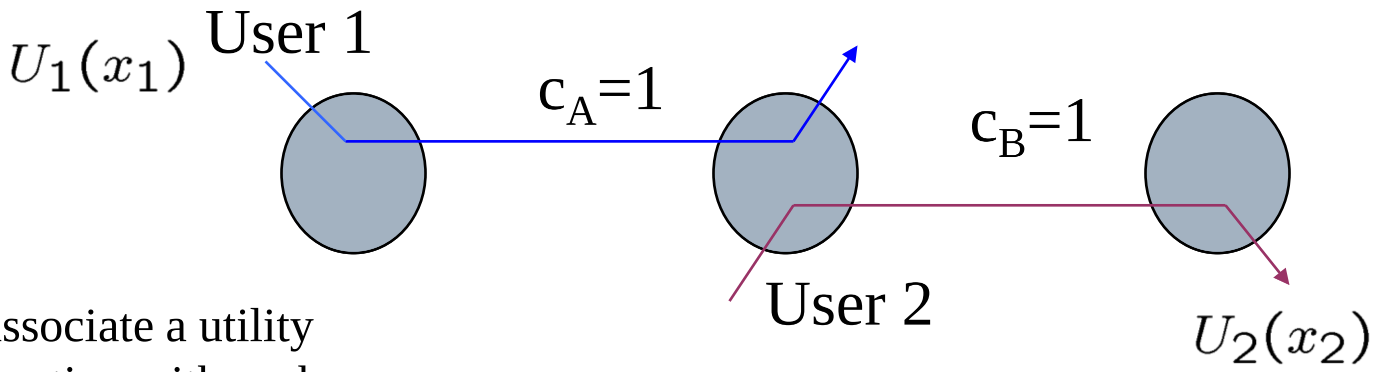
- Links A and B can serve one packet in each time instant
- Both links cannot be active simultaneously: interference constraint
- Two users:
  - User 1 traverses link A only
  - User 2 traverses link B only
- How should we divide the capacity of the two links between the two users while respecting the interference constraint?

What is Resource Allocation?



- Determine the appropriate values for these variables
  - x_1: rate at which user 1 is allowed to transmit data
  - x_2: rate at which user 2 is allowed to transmit data
  - μ_a: fraction of time link a is active
  - μ_b: fraction of time link b is active

2-Link, 2-User wireless network



Either link A or link B can be active, but not both.

(associate a utility function with each user)

$$\max_{x,\mu \geq 0} \sum_i U_i(x_i)$$

Constraints:

$$x_1 \leq \mu_a$$

$$x_2 \leq \mu_b$$

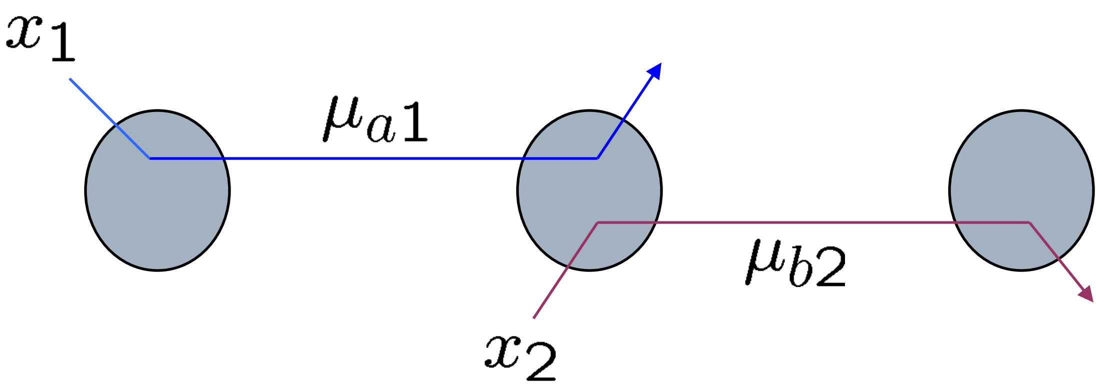
$$\mu_a + \mu_b \leq 1$$

Flow conservation constraint for at Link 1:

x_1 is the arrival rate of user 1

μ_a is the fraction of time link A is activated

Lagrange Multipliers



$$\max_{x,\mu} \sum_i U_i(x_i) - p_1(x_1 - \mu_a) - p_2(x_2 - \mu_b)$$

subject to

$$\mu_a + \mu_b \leq 1$$
$$x, \mu \geq 0$$

Lagrangian Decomposition

Congestion control:

$$\max_{x \geq 0} \sum_i U_i(x_i) - p_1 x_1 - p_2 x_2$$

$\Rightarrow$ User 1: $\max_{x_1 \geq 0} U_1(x_1) - p_1 x_1$

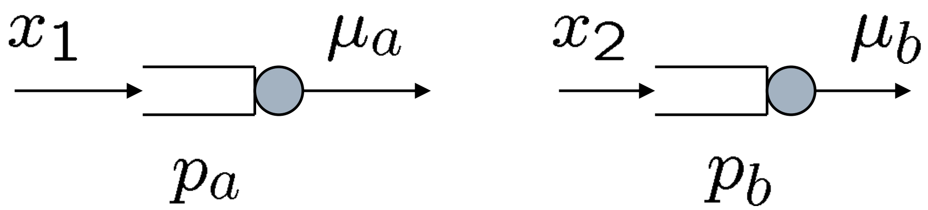
…

MaxWeight Algorithm for Scheduling:

$$\max_{\sum \mu_i \leq 1} \mu_a p_1 + \mu_b p_2$$

Solution is an extreme point!
Only one link activated at a time

Resource Constraints and Queue Dynamics



$$\max_{x,\mu \geq 0} \sum_i U_i(x_i)$$

subject to

$$\begin{aligned} x_1 &\leq \mu_a \\ \dot{p}_1 &= x_1 - \mu_a \\ x_2 &\leq \mu_b \\ \dot{p}_2 &= x_2 - \mu_b \end{aligned}$$

- Lagrange multipliers = Queue lengths

Recap: Queueing and Optimization

- Each constraint is represented by a queue:

$$y \leq x$$

- Stability of the queue implies constraint is satisfied and vice-versa; resource allocation is some form of the Maxweight algorithm with queue lengths as weights
  - Dual formulation reveals the form of the MaxWeight algorithm (Tassiulas-Ephremides, 1992)
- Queue length proportional to the Lagrange multiplier (stochastic arrivals/departures , ϵ: step-size parameter):

$$q(k+1)=[q(k)+\epsilon\,(Y(k)-X(k))]^+$$

Typical Theorem

- Let
  - J^* be the optimal value of the objective of the deterministic problem
  - J_{st} be the long-run average objective in the real system, which is usually stochastic (stochastic arrivals, stochastic channels, etc.)
- Theorem: The queues are stable. Further,

$$\mathbb{E}(J_{st}) \geq J^* - K\epsilon; \mathbb{E}(\sum_l q_l) \leq f(1/\epsilon)$$

- Eryilmaz & Srikant (2005); Neely, Modiano, Li (2005); Stolyar (2005); Decomposition also by Lin & Shroff (2004)

Issues

- All constraints formulated in terms of long-term averages

- Does this mean only long-lived elastic flows can be modeled using this framework?

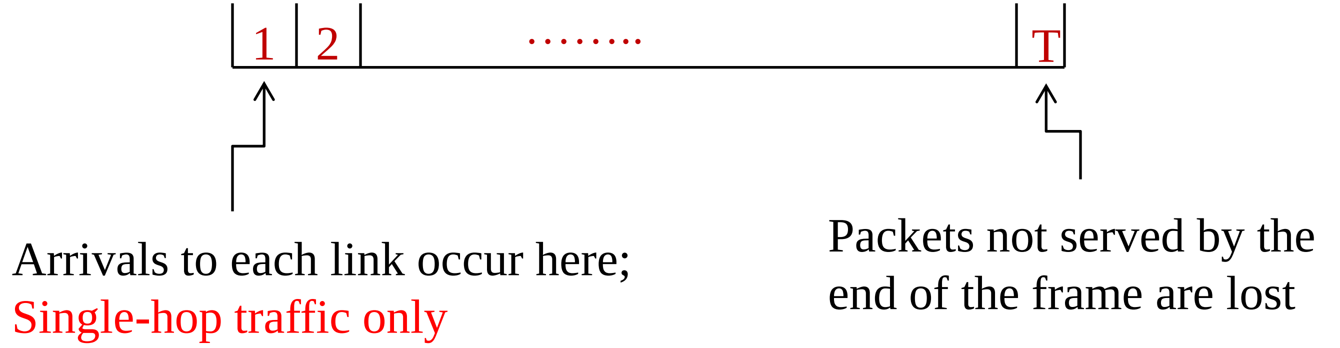
- We will present two applications which can be modeled using this framework:
  - Packets with deadlines: constraint in terms of lower bounds on the long-run fraction of packets delivered before deadline expiry, i.e., a certain % of packets have to served before deadline expires

  - A mixture of long-lived and short-lived flows: Short-lived flows bring a finite number of packets to the network and depart when their packets are delivered.

Application I: Per-packet Deadlines

- Consider an ad hoc network consisting of L links

- Time is divided into frames of T slots each (Hou, Borkar, Kumar, '09)



- QoS requirement for link l: fraction of packets lost due to deadline expiry has to be less than or equal to p_l

Schedule (Matrix) for Each Frame

	Time Slot 1	Time Slot 2	.	.	Time Slot T
Link 1	1 (ON)	0	0	1	1
Link 2	1	0	1	0	0
.	0 (OFF)	1	0	0	1
.	0	1	0	0	1
Link L	0	1	1	0	0

- In each time slot, select a set of links to be ON, while satisfying some interference constraints
- Thus, a schedule is an LxT matrix of 1s and 0s

Problem: Find a schedule in each frame such that the QoS constraints are satisfied for each link

An Optimization Formulation

- S_{lk} = 1 if link l is scheduled in time slot k
- A_l: Number of arrivals to link l in a frame, a random variable, with mean λ_l (unknown)
- Constraint: Average number of slots allocated must be greater than or equal to the QoS requirement for each link l

$$E[\min(\sum_k S_{lk}, A_l)] \geq \lambda_l(1-p_l)$$

- A dummy optimization problem (B is some constant):

$$\max B$$

Fictitious Queue

- Recall $x \geq y$ corresponds to

- Similarly,

$$E[\min(\sum_k S_{lk}, A_l)] \geq \lambda_l(1-p_l)$$

corresponds to

Upon each packet arrival to link l, add a packet to this queue with prob. $(1-p_l)$

Remove packet from the queue every time a packet is successfully scheduled

Deficit counter: Keeps track of deficit in QoS

Optimal Schedule

- d_l: deficit of link l
- Choose a schedule at each frame to maximize

$$\sum_l d_l \left(\sum_k S_{lk}\right)$$

slots allocated to link l

subject to $\sum_k S_{lk} \leq A_l$

- This is simply the MaxWeight algorithm where the deficits are used as weights, instead of real queue lengths
- The constraint simply states that the number of slots allocated to link l in a frame should not be greater than the number of arrivals in the frame

Resource Allocation

- Beyond just meeting constraints: allocate extra resources to meet some fairness constraint

$$\max \sum_l w_l \left(\sum_k S_{lk}\right)$$

subject to $$E\left[\min\left(\sum_k S_{lk}, A_l\right)\right] \geq \lambda_l(1-p_l)$$

- Optimal Solution becomes obvious after adding constraint to the objective using Lagrange multipliers: Choose schedule S in each frame to maximize

$$\sum_l (w_l+\epsilon\, d_l)\left(\sum_k S_{lk}\right)$$

Theorem

- Result 1:

$$E(w_l\, x_{li}) - \sum_l w_l\, x_{li}^* = O(\epsilon)$$

- Result 2:

$$E(\sum_l d_l) = O(1/\epsilon)$$

ϵ provides a tradeoff between optimality and queue lengths and deficits

Application II: Downlink Scheduling

- Model: A Base station transmitting to a number of receivers
- The base station can transmit to only one user at a time
- Classical Model: a fixed number of users, say N
- Each user's channel can be in one of many states:
  - $R_i(t)$: Rate at which the base station can transmit to User i if it chooses to schedule user i
- Classical problem (channel states are known to the base station): Which user should the base station select for transmission at each time instant?

Classical Solution

- Suppose that the goal is to maximize network throughput:
  - i.e., the queues in the network must be stable as long as the arrival rates lie within the capacity region of the system

- (Tassiulas-Ephremides '92): Transmit to user i such that

$$i \in \arg\max_j q_j(t) R_j(t)$$

- Solution can be derived from optimization considerations as mentioned earlier in the case of ad hoc networks
  - One has to simply account for the time-variations in the channel

New Model: Short-lived Flows

- What if the number of flows in the network is not fixed?
  - Each flow arrives with a finite number of bits. Departs when all of its bits are served
  - Flows arrive according to some stochastic process (Poisson, Bernoulli, etc.)

- The number of bits in each flow is finite, so need a different notion of stability since queues cannot become large
  - Need the number of flows in the system to be "finite"

Van de Ven, Borst, Shneer '09: The MaxWeight algorithm need not be stabilizing; the number of flows can become infinite even when the load lies within the capacity region

Necessary condition for stability

- Suppose each channel has a maximum rate $\mathbb{R}^{max}$

- A necessary condition for stability:
  - F: File size, a random variable. Expected number of time slots (workload) required to serve a file is

$$E(\lceil F/\mathbb{R}^{max} \rceil),$$

    achieved when each user transmits only when its channel is in the best condition

  - λ: Rate of flow arrivals (number of flows per time slot)

Necessary condition for stability : $\lambda \; E(\lceil F/\mathbb{R}^{max} \rceil) \leq 1$

Scheduling Algorithm

- Transmit to the user with the best rate at each time instant, $\text{Max}_i\ R_i(t)$

- Does not even consider queue lengths in making scheduling decisions

- Why does it work?
  - When the number of flows in the network is large, some flow must have a rate equal to R^{max} with high probability
  - Thus, we schedule users when their channel condition is the best; therefore, we use the minimum number of time slots to serve a user

Short-Lived and Long-Lived Flows

- Now consider the situation where there are some long-lived (persistent) flows in the networks
- For simplicity, we will consider the case of one long-lived flow which generates packets at rate ν packets per time slot
- Solution: using an optimization formulation

Capacity constraints

- R_c: rate at which the long-lived flow can be served when its channel state is c (a random variable)
- π_c: probability that the long-lived channel state is c
- p_c: probability of serving the long-flow in state c
- Constraints:
  - Long-lived flows: $\nu \leq \sum_c \pi_c \, p_c \, R_c$
  - Short-lived flows: $\lambda \, E(\lceil F/R^{max} \rceil) \leq \sum_c \pi_c \, (1-p_c)$

Optimization Interpretation

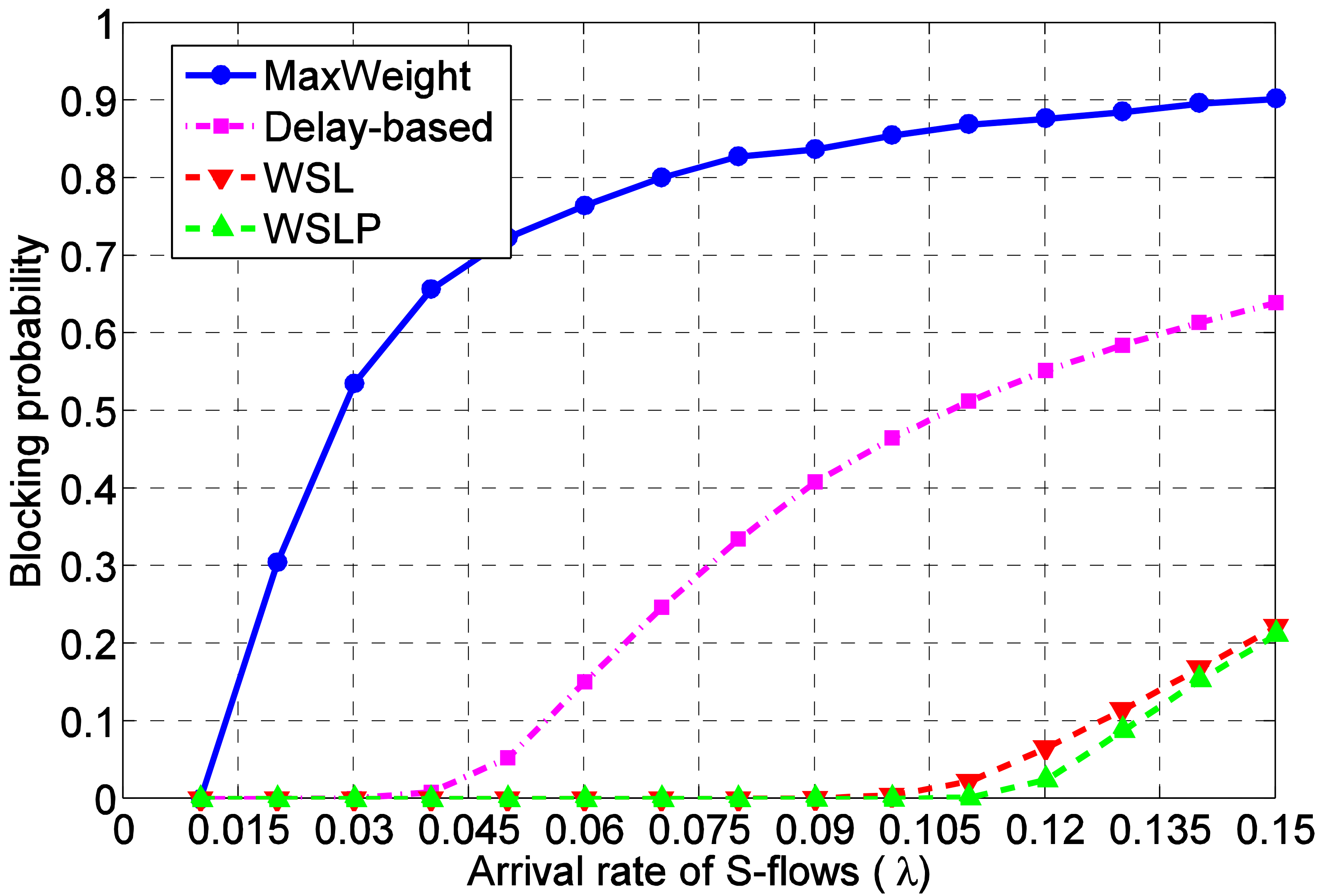
- Lagrange multiplier of $\nu \leq \sum_c \pi_c p_c R_c$
  - Left-hand side is packet arrival rate, right hand side is packet departure rate of long-lived flows
  - So the Lagrange multiplier is (proportional to) the queue length of the long-lived flow

- Lagrange multiplier of $\lambda E(\lceil F/R^{max} \rceil) \leq \sum_c \pi_c (1-p_c)$
  - Left-hand side is the minimum number of slots (workload) required to serve short-lived flows, the right-hand side is the number of slots available
  - So, the Lagrange multiplier is (proportional to) the minimum number of slots required (workload) to serve the short-lived flows in the solution

Optimization Solution

- If the workload of short-lived flows is larger than the queue length of the long-lived flow, then serve a short-lived flow
  - Choose the flow with the best channel condition
- Else, serve the long-lived flow
- Extensions:
  - More than one long-lived flow
  - Different short-lived flows have different $\mathbb{R}^{max}$
  - The $\mathbb{R}^{max}$'s are unknown; learn them, by using the best channel condition seen by each flow so far

Simulations



Conclusions

- Optimization theory provides a cookbook for solving resource allocation problems in communication networks

- Lagrange multipliers are proportional to queue lengths
  - May need to interpret "queue length" appropriately: e.g., deficit counter, workload

- Resource allocation decisions are made by comparing Lagrange multipliers using the MaxWeight algorithm
  - Typically obvious when writing out the dual formulation

- Tradeoff between optimality and queue lengths using the drift of Lyapunov functions