The Prices of Packets: End-to-end delay Guarantees in Unreliable Networks

Rahul Singh
LIDS
MIT

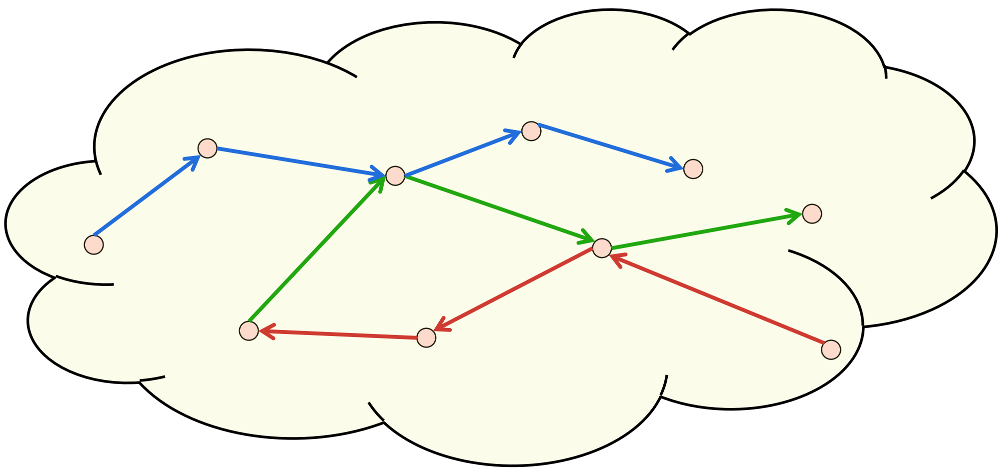
P. R. Kumar
Dept. of Electrical and Computer Engineering
Texas A&M University

Rahul Singh

Email: prk.tamu@gmail.com
Web: http://cesg.tamu.edu/faculty/p-r-kumar/

4th MobiHoc Workshop on the Frontiers of Networks: Theory and Algorithms, MobiHoc 2016
Paderborn
July 5, 2016

Unreliable multi-hop networks with end-to-end delay constraints

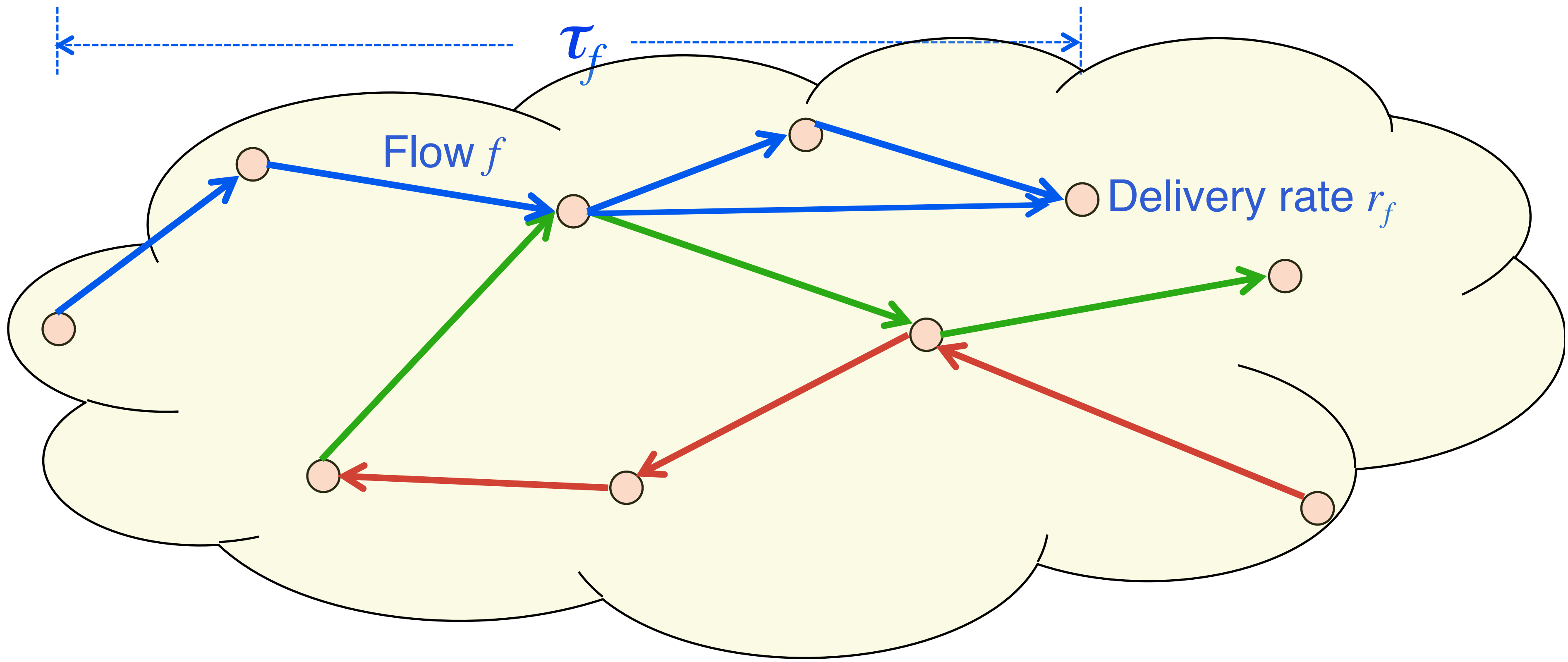


- Channels are unreliable
- Flows have throughput requirements
- Packets have end-to-end deadlines

} Timely throughput

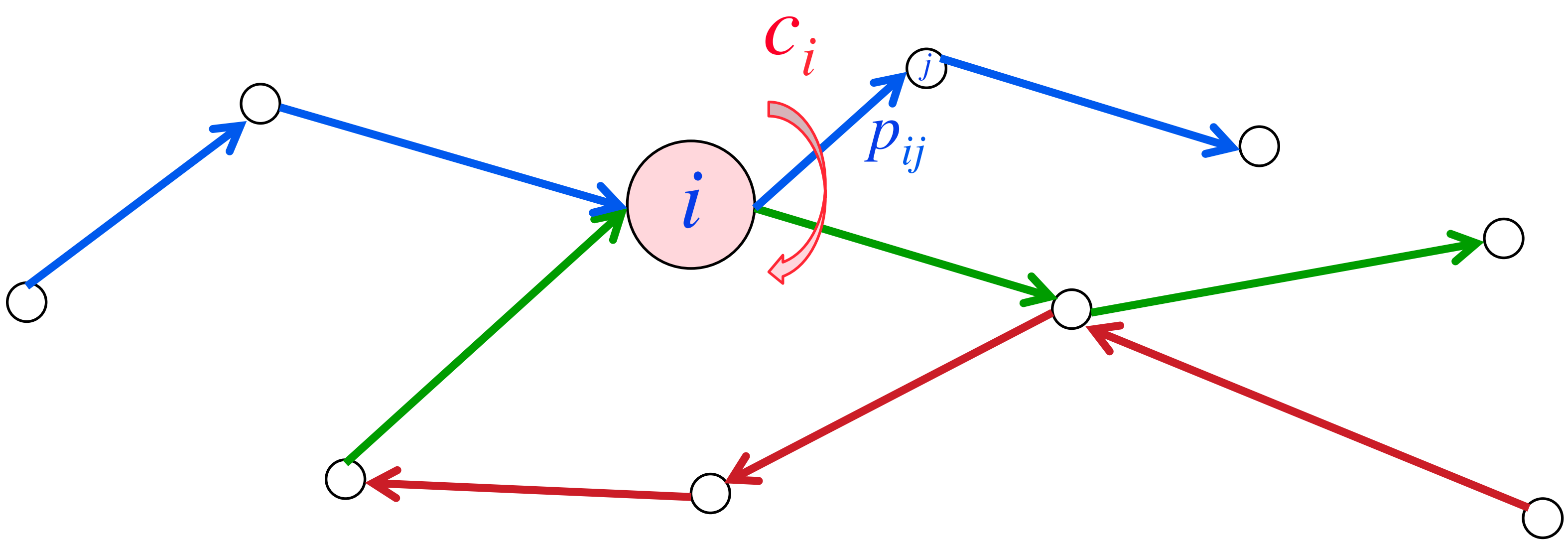
How to schedule such multi-hop networks?

Multi-hop network



- F flows
- Flow f has an end-to-end deadline τ_f
- Flow f needs a packet delivery rate of r_f

Nodal constraints

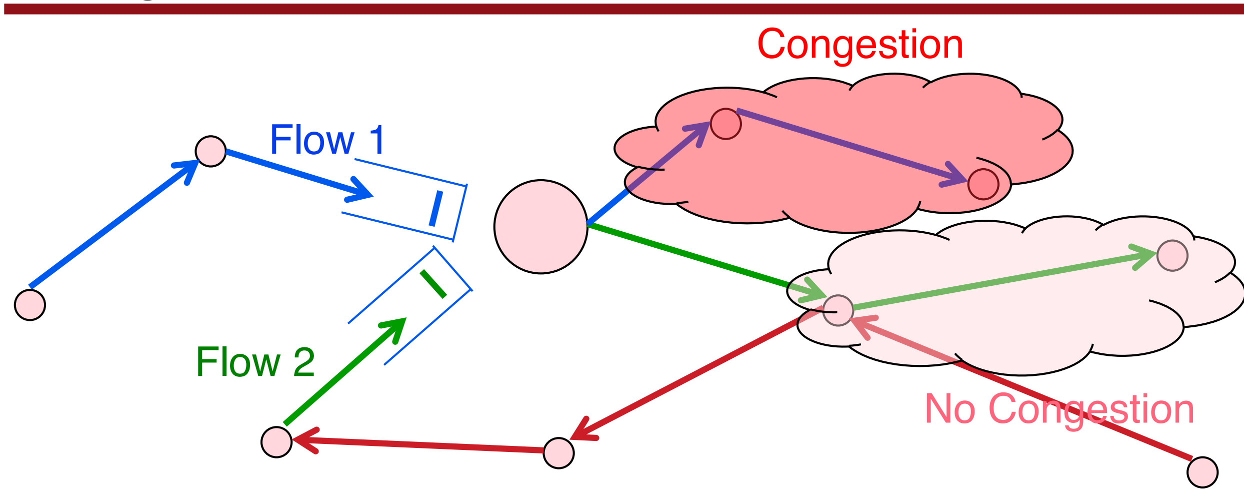


- Packet transmission succeeds with probability p_{ij}
- Node i has an average power constraint

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left(\#\text{ of packets transmitted by node } i \text{ at time } t\right)\le c_i$$

- Neglect interference

Challenge of scheduling a distributed system



Transmit packet from Flow 2 since Flow 1 has downstream congestion

- Optimal scheduling requires knowledge of the complete network state
- Obtaining network state instantaneously itself requires solving end-to-end delay problem!
- Optimally scheduling a distributed system is difficult!
- Even if each node could obtain complete state, DP is intractable
  - Huge state space: $(V\Delta)^{F\Delta}$

Objective

$$Max \sum_f \alpha_f r_f$$

Where: r_f = Throughput of packets of flow f

that have an end-to-end delay $\leq \tau_f$

= Timely throughput of flow f

- The timely throughput of flow f is weighted by α_f
- How to schedule the network?

Solution

- Constrained optimization problem over stationary randomized policies π

$$\underset{\pi}{Max} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{f} \alpha_f (\#\text{ of packets of flow } f \text{ delivered in time at time } t)$$

$$\text{Subject to} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(\#\text{ of packets transmitted by node } i \text{ at time } t\right) \le c_i$$

- Lagrangian $\mathcal{L}(\pi,\lambda)$

$$\underset{\pi}{Max} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left\{ \sum_{f} \alpha_f (\#\text{ of packets of flow } f \text{ delivered in time at time } t) - \sum_{t=1}^{T} \lambda_i \left(\#\text{ of packets transmitted by node } i \text{ at time } t\right) \right\} + \sum_i \lambda_i c_i$$

- Packet-by-Packet Decoupling

$$\underset{\pi}{Max} \limsup_{T\to\infty} \frac{1}{T} \sum_{f} \sum_{\text{Packets of flow } f \text{ released before time } T} \left\{ \alpha_f 1(\text{Packet is delivered on time}) - \sum_i \lambda_i 1(\text{Packet is transmitted by Node } i) \right\}$$

Single Packet's Costs and Reward

$$\alpha_f 1(\text{Packet is delivered on time})$$
$$-\sum_i \lambda_i 1(\text{Packet is transmitted by Node } i)$$

- Single packet of flow f receives reward α_f if it reaches its destination in time
- Pays λ_i to node i whenever it requests transmission at node i

Single Packet Optimal Transportation Problem

- State of packet = (i, s)
  = (Node where it is, Time-till-deadline)

- Dynamic Programming equation

$$
\begin{aligned}
V^f(i,s) &= \max\{V^f(i,s-1), \\
&\max_{j:(i,j)\in\mathcal{E},E}\{-\lambda_i + p_{(i,j)}(E)V^f(j,s-1) \\
&+\left(1-p_{(i,j)}(E)\right)V^f(i,\left(s-1\right)^+)\}\}, \\
V^f(d_f,s) &= \beta_f \text{ if } s \geq 0.
\end{aligned}
$$

- Easy to solve: Only $V\Delta$ states

Optimal distributed solution

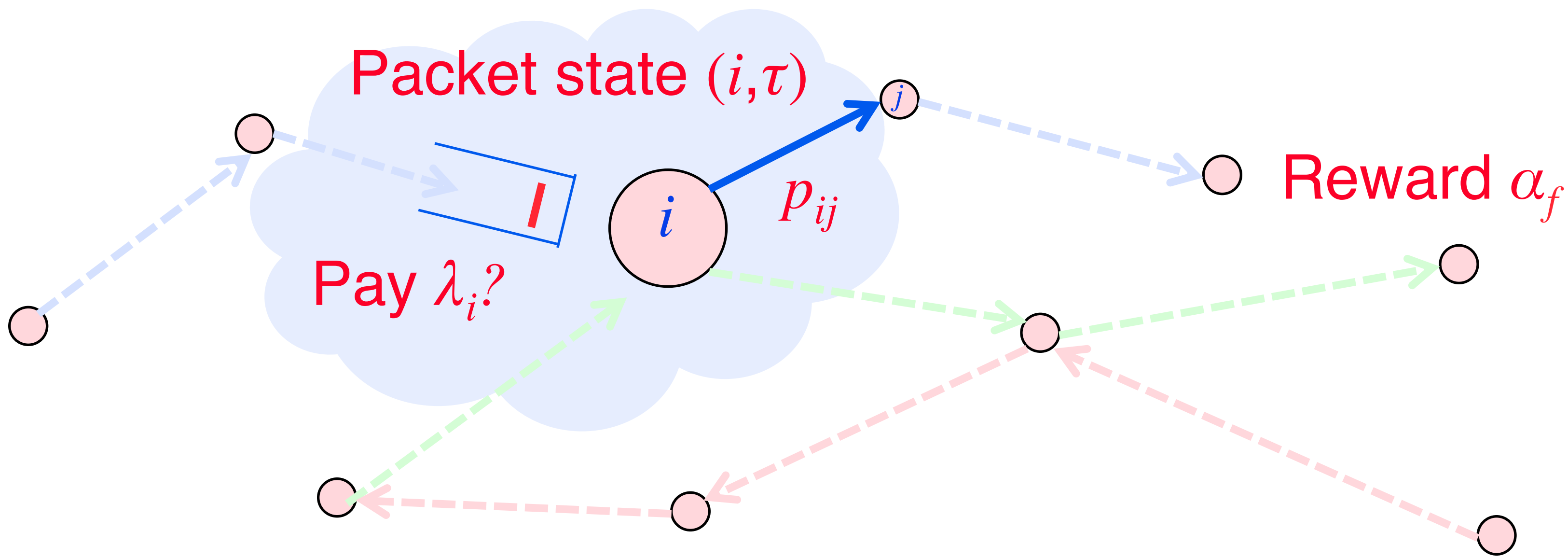
- The optimal solution completely decouples!

- Each packet makes decision to be transmitted or not, depending only on its *own* state (*location, time to deadline*)

- Optimal scheduling of a packet does *not* depend on
  - State of other nodes
  - State of other flows
  - Even other packets within its own flow!

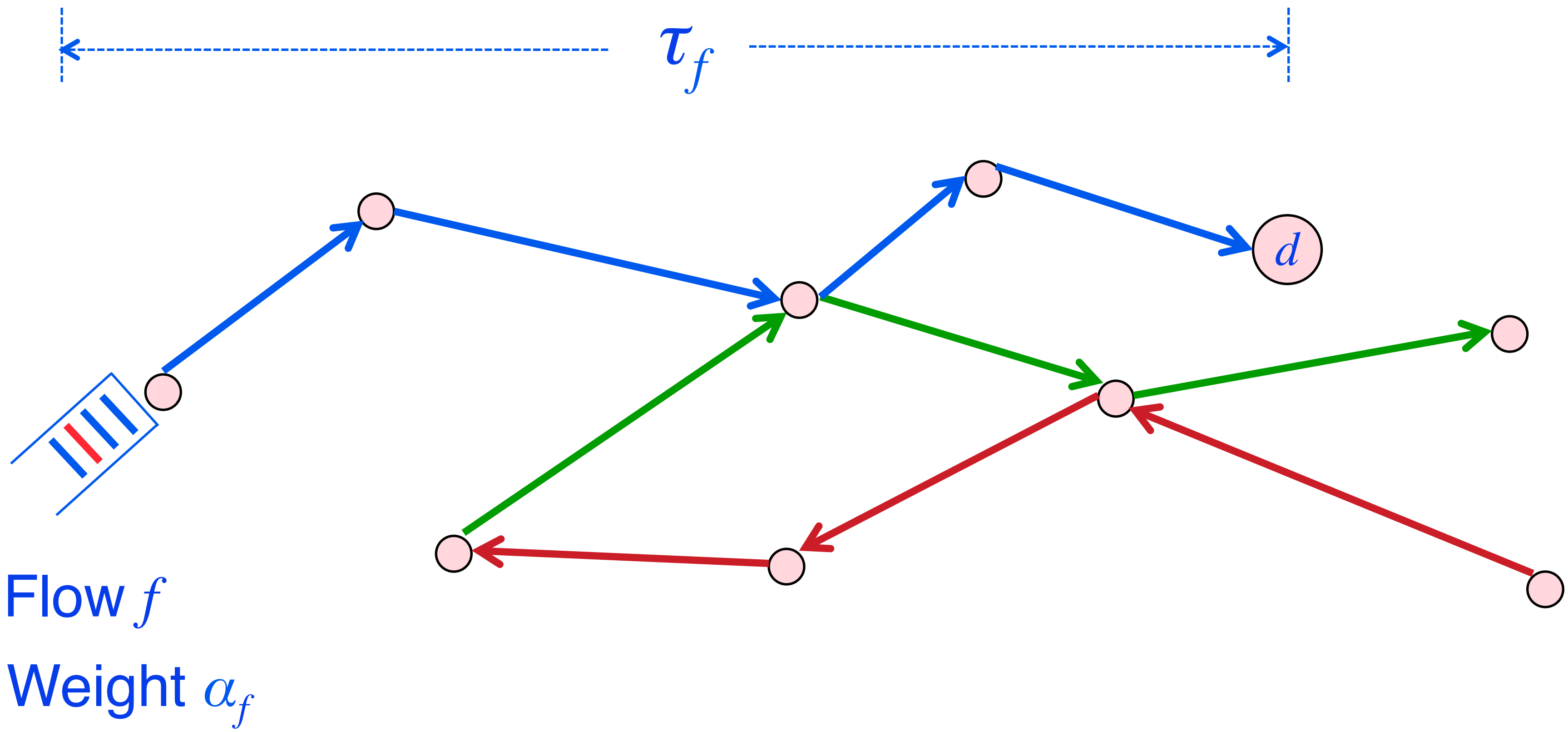
Packet decoupled solution

- Each packet decides at each time t whether to be transmitted or not

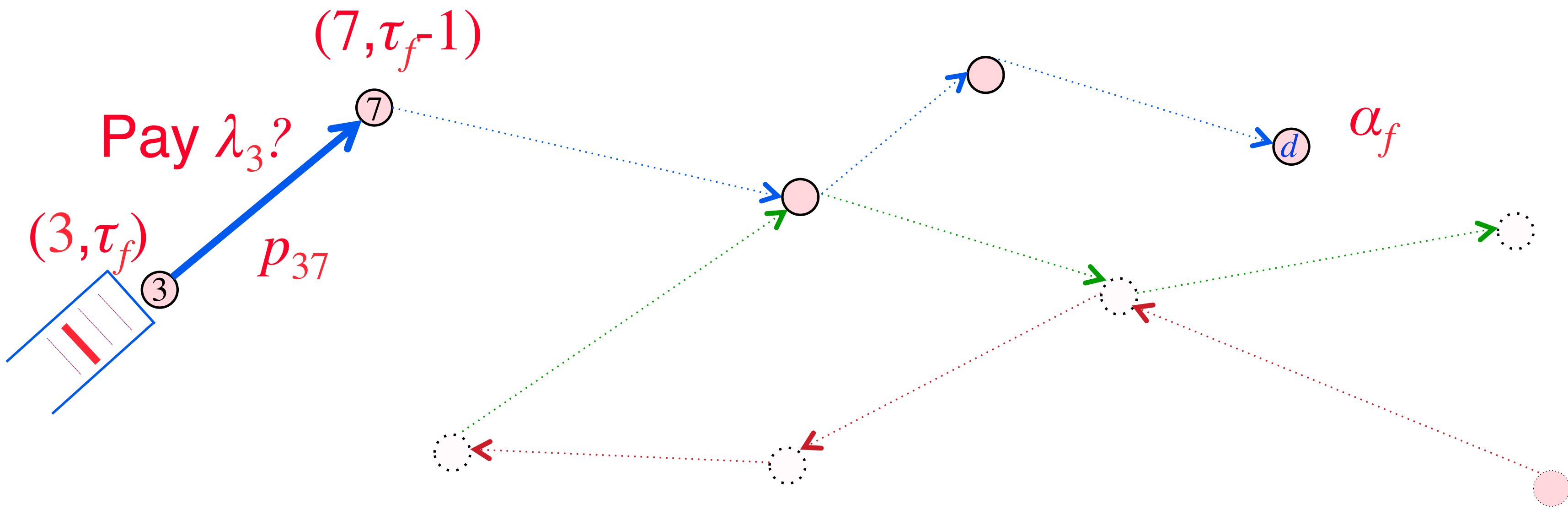


- Packet collects a reward α_f if it reaches its destination before its deadline
- Packet pays node i a price λ_i for being transmitted
- Packet level decision making: Should it pay price and be transmitted?

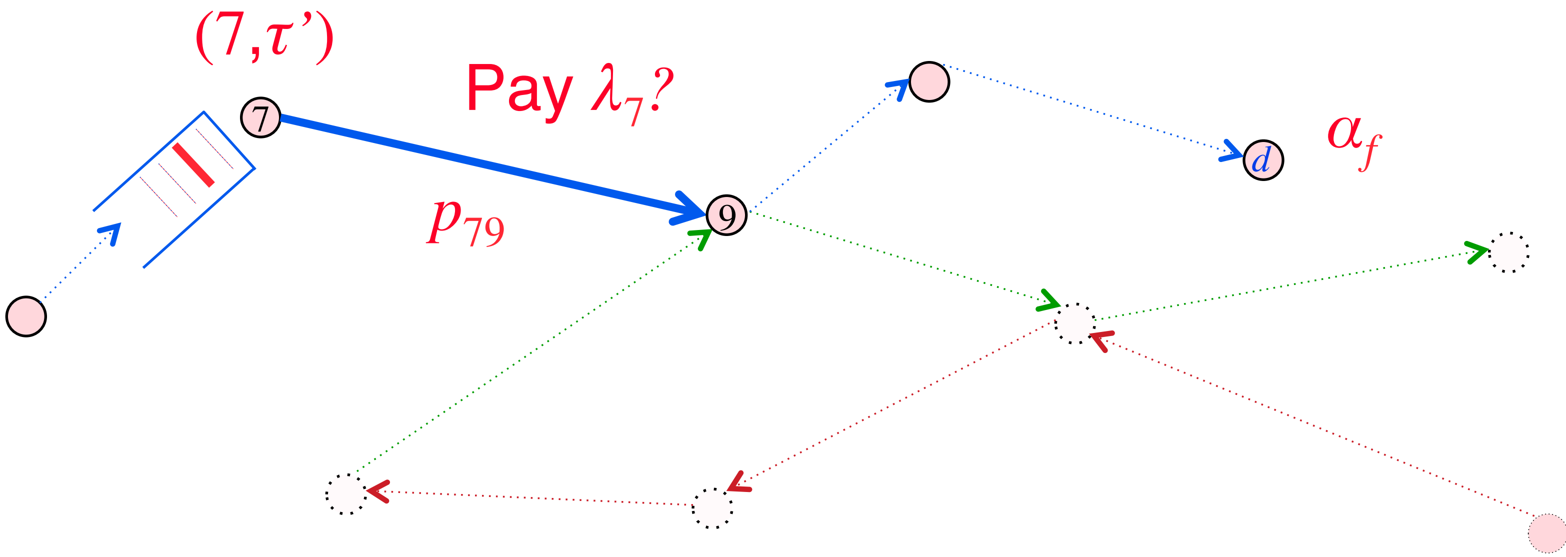
Optimal packet decoupling



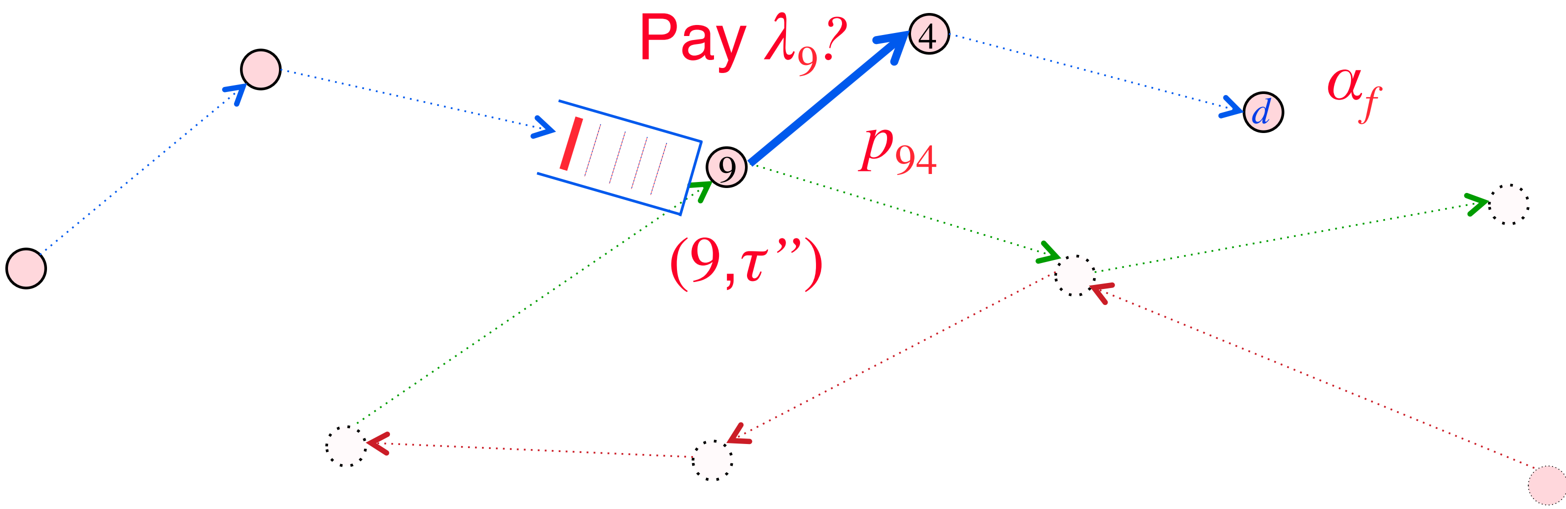
Optimal packet decoupling



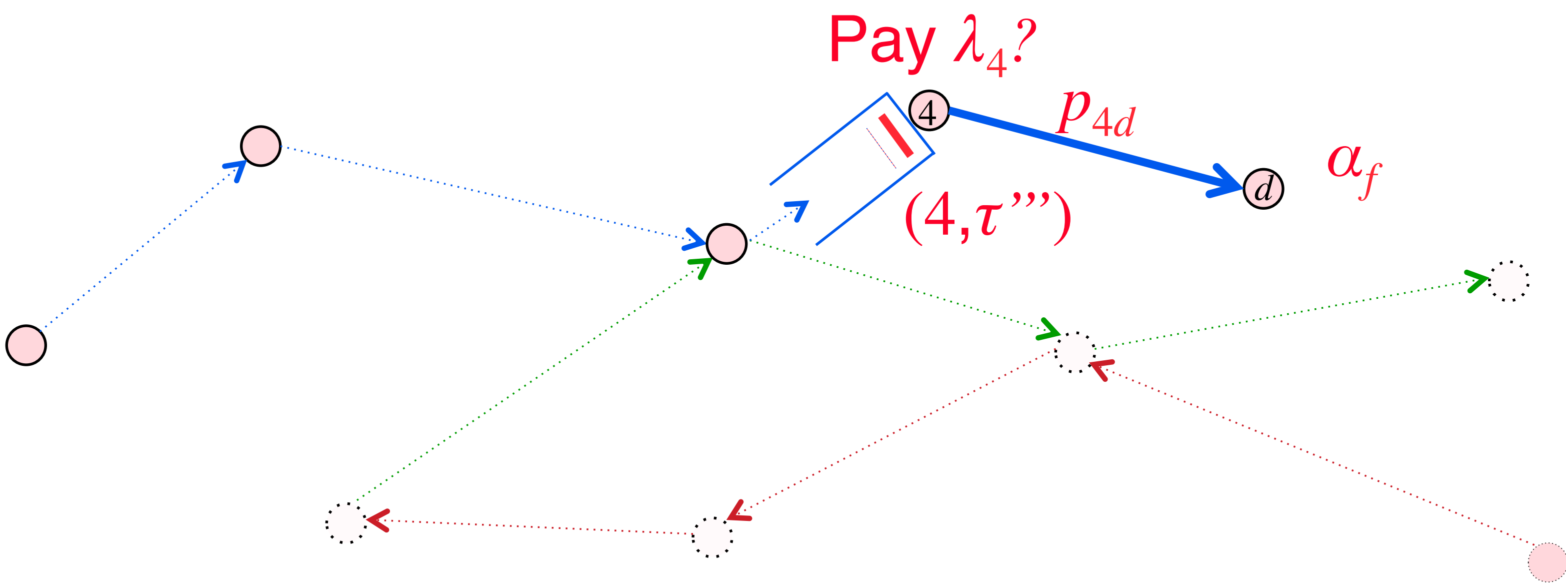
Optimal packet decoupling



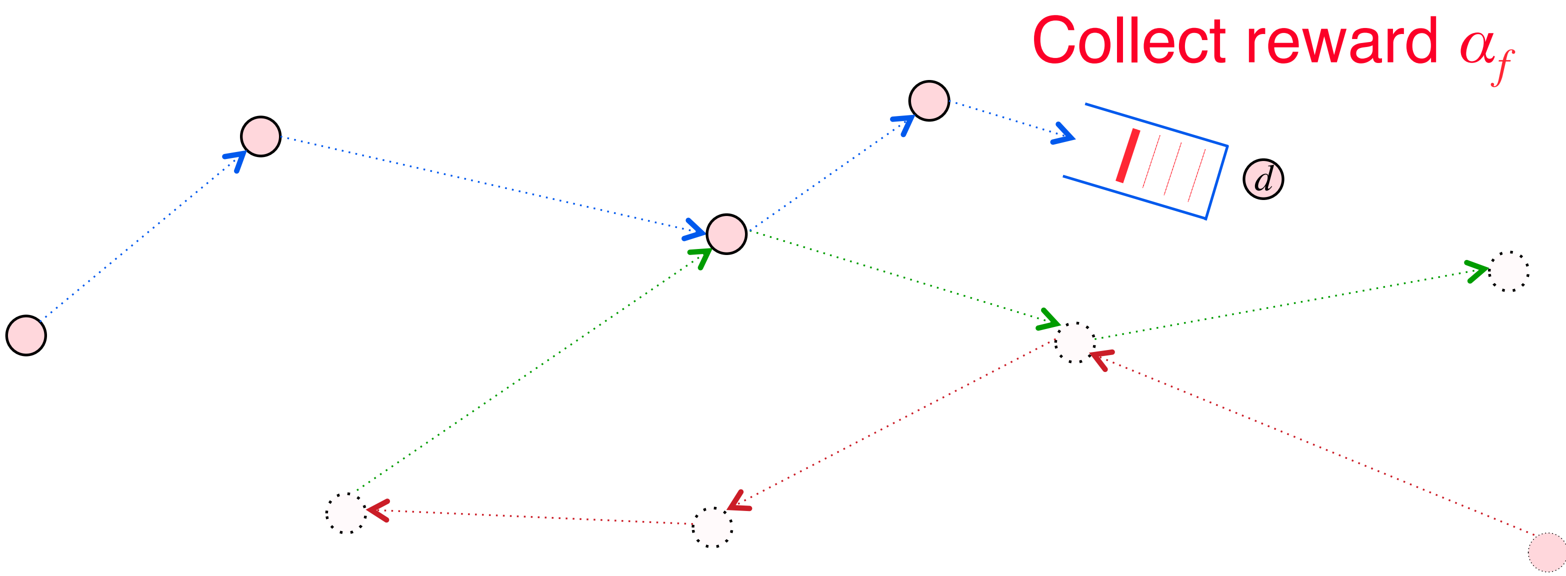
Optimal packet decoupling



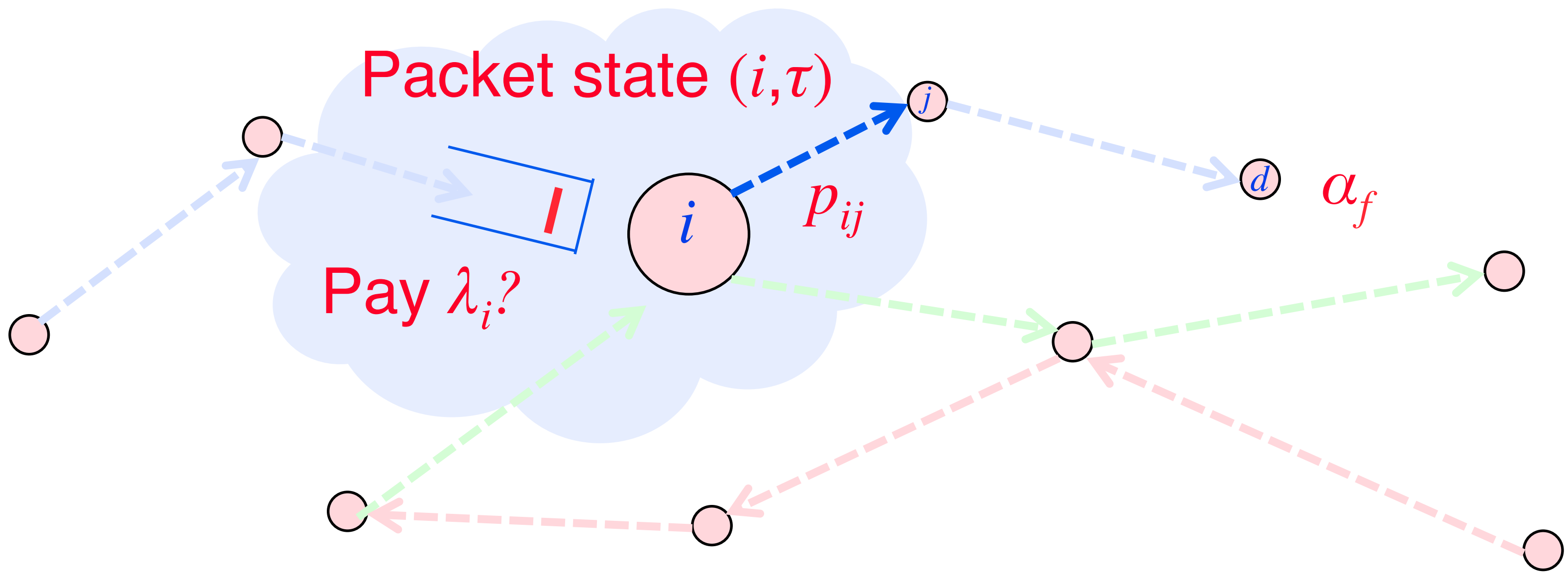
Optimal packet decoupling



Optimal packet decoupling



Packet level decision making



$$V(d,t) = \alpha_f \text{ for all } t \geq 0$$

$$V(i,\tau) = Min\left\{\lambda_i + p_{i,j}V(j,\tau-1) + (1-p_{i,j})V(i,\tau-1),\ V(i,\tau-1)\right\}$$

- Packet solves Dynamic Program offline
- Packet state space size is $V\Delta$

How to obtain prices?

- If price λ_i is too low
  - Too many packets ask to be transmitted
  - Average power λ_i constraint is exceeded
- If price λ_i is too high
  - Too few packets ask to be transmitted
  - Average power available λ_i is not used
- Suggests tatonnement

$$\lambda_i^{n+1} = \lambda_i^n + \epsilon\ [\text{Power consumed by node } i - c_i].$$

- But even if price is exactly right, we will need to randomize some flow's decisions to get the power to be exactly used up

Dual Problem

- The Dual function is $D(\lambda) = \max_{\pi} \mathcal{L}(\pi, \lambda)$
- "Max" is attained by Single Packet Transportation Problem
- Dual Problem is $\max_{\lambda \geq 0} D(\lambda)$
- No Duality Gap, since can be reduced to LP

Optimality condition

- Suppose λ^* is price vector
- $\pi(\lambda^*)$ optimal randomized policy for single-packet transportation problem for each flow f
- Suppose at every node i,
  - Either power constraint is satisfied with equality by $\pi(\lambda^*)$

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left(\#\text{ of packets transmitted by node } i \text{ at time } t\right) = c_i$$

  - Or $\lambda_i^*=0$
- Then $\pi(\lambda^*)$ and λ^* are optimal by Complementary Slackness

Combine Single-Packet Transportation Problems of all Flows

$$\max \sum_{f\in F}\sum_{s=0}^{\tau_f}\sum_{i\in V} A_f \xi^f(i,d_f,s)\, p_{i,d_f}$$ Reward

Use state-action probabilities

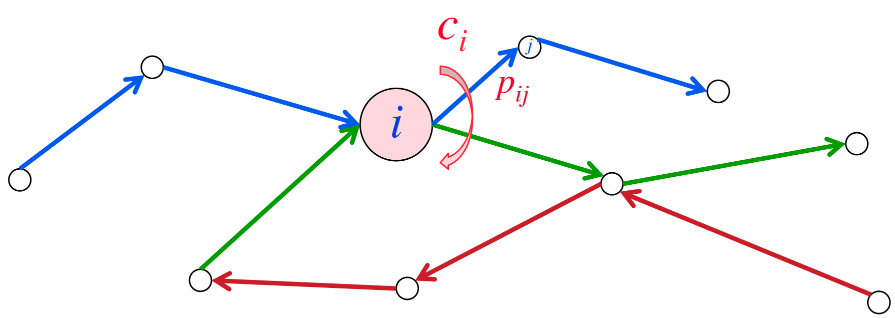
$$\sum_{f\in F}\sum_{s=0}^{\tau_f}\sum_{j\neq i} A_f \xi^f(i,j,s) E \le P_i \quad \forall i \in V,$$ Power Constraint

$$\sum_{j\in V, j\neq d_f} \xi^f(j,i,s)\, p_{j,i} + \sum_{m\in V} \xi^f(i,m,s)(1-p_{i,m}) = \sum_{k\in V} \xi^f(i,k,s-1) \quad \forall i \neq d_f, 1 \le s \le \tau_f$$ Balance equations

$$\sum_{j\in V} \xi^f(s_f,j,0) = 1 \quad \forall f, \qquad \xi^f(i,j,s) \ge 0$$ Probabilities

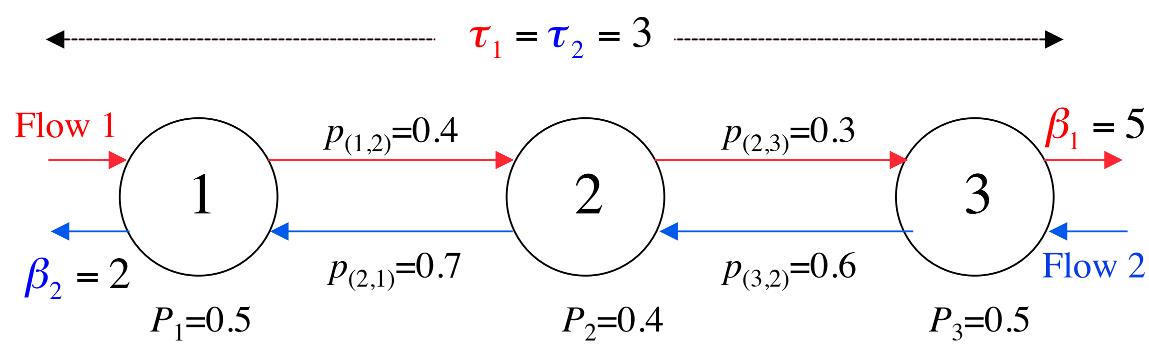
- Very tractable LP solution: Low complexity
- Just $|V|^2 F\Delta$ variables, $|V| + |V| F\Delta + F + |V|^2 F\Delta$ constraints
- Reduction from exponential $(V\Delta)^{F\Delta}$ complexity

Near Optimality for Peak Power Constraint



- Suppose Node i can only transmit c_i packets concurrently
- Simply truncate at c_i
  - Similar to Whittle's relaxation for restless bandits
- **Theorem**
- Policy is asymptotically $O\left(\frac{1}{\sqrt{N}}\right)$ optimal as the total network capacity is scaled by N
- Quantitative approach to end-to-end deadline scheduling

Example: Explicit solution



- Optimal solution
- Optimal prices

$$\lambda = (0.068, 1.4, 0)$$

- Optimal solution

$$\left.\begin{aligned} \pi^1(1,3) = 0.5, \pi^1(1,2) = 0, \\ \pi^1(2,2) = 1, \pi^1(2,1) = 1, \\ \pi^2(3,3) = 1/13, \pi^2(3,2) = 0, \\ \pi^2(2,2) = 1, \pi^2(2,1) = 1. \end{aligned}\right\}$$

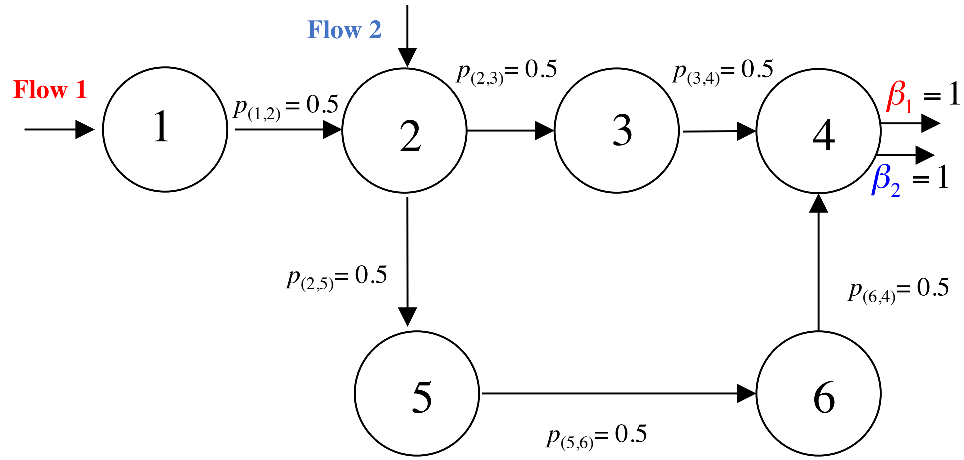
Flow 1:
Node 1 transmit with probability 0.5 if time-till-deadline is 3, else drop

Flow 2:
Node 3 transmit with probability 1/13 if time-till-deadline is 3, else drop

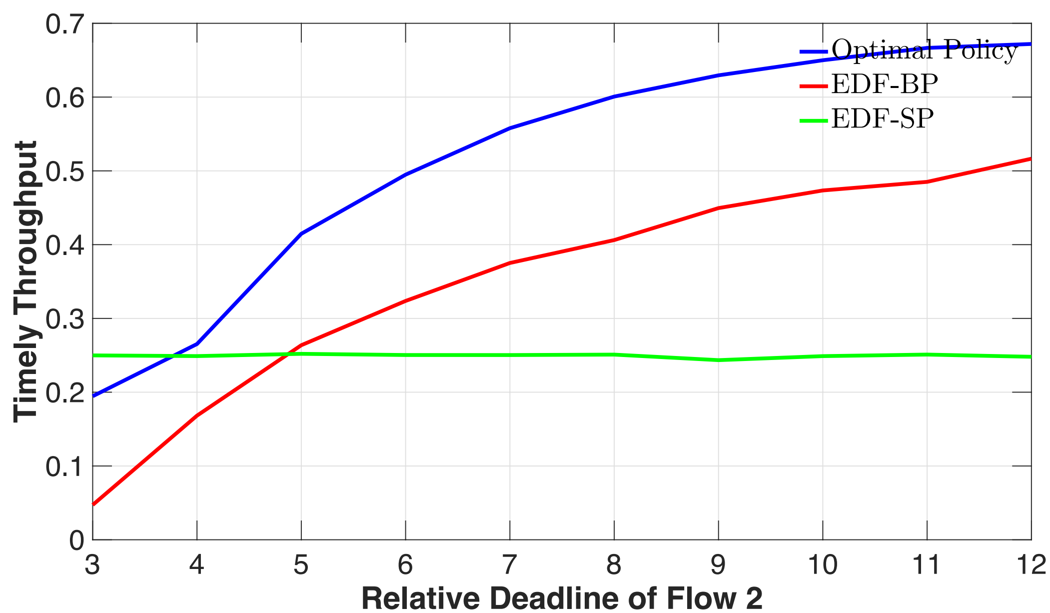
Node 2: Both Flows Transmit

Example: Numerical Computation and Simulation Comparison

- Comparison with EDF-BP and EDF-SP



- $A_1 = A_2 = 1$
- $C_{(i,j)} = 1$
- $\Delta_1 = \Delta_2 + 1$



Concluding remarks

- A quantitative optimal distributed solution for end-to-end delay-based wireless networking for unreliable links
  - Deadlines
  - Throughput
  - Unreliable wireless channels
  - Multi-hop wireless networks
- Issues not considered here: Contention, interference, coding
- Lots of scope for systems work

Thank you