Distributed Control and Quality-of-Service in Multihop Networks

Ashok Krishnan K.S.

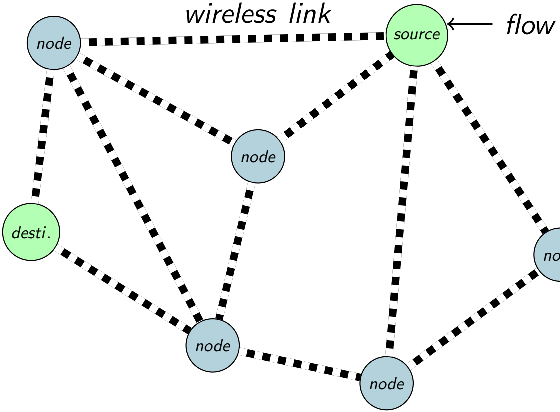
(joint work with Prof. Vinod Sharma)

Performance Analysis Lab, Department of ECE, IISc

Introduction

- Multihop Wireless Networks
- Distributed Control
- Quality of Service (QoS) Guarantees
- Complexity and Performance

A Multihop Wireless Network



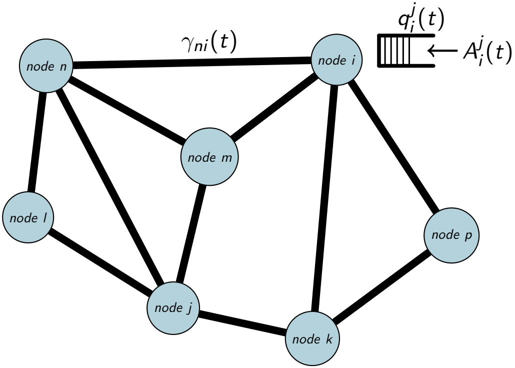
Control of the Network

- Various parameters
- Optimization
- Feasibility
- Distributed?

Quality of Service (QoS)

- Mean Delay
- Hard Deadline
- Rate

System Model



A Wireless Network

System Model

- Slotted System, $t \in \{0, 1, 2, ...\}$
- Arrival Process $A_i^c(t)$ with mean λ_i^c
- QoS for flows
  - End to End Mean Delay Guarantee
  - End to End Hard Deadline Guarantee

System Model

- Power vector $p(t) = [p_{ij}(t)]_{1 \leq i,j \leq N} \in \mathcal{P}$
- Channel Rate $r_{ij}(t) = f(p(t), \gamma(t))$
- $r_{ij}(t) = \log_2 \left(1 + \frac{p_{ij}(t)\gamma_{ij}(t)}{\mathcal{N}_j(t) + \sum_{k \neq i} \sum_{l \in V} p_{kl}(t)\gamma_{kl}(t)} \right)$
- $q_i^c(t+1) = q_i^c(t) - \mu_{OUT,i}^c(t) + \mu_{IN,i}^c(t) + A_i^c(t)$

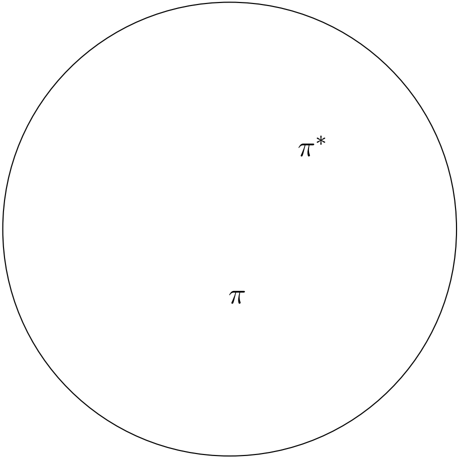
Control

- A Control Policy $\{\pi(t)\}_{t=0,1,2,\ldots}$
- $\pi = g(\gamma, Q)$
- π optimizes some cost function

Related Work

- Throughput Optimal Policies
  - Tassiulas and Ephremides (1992)
  - Neely, Modiano and Rohrs (2005)
- Non SINR models
  - Bui, Eryilmaz, Srikant, and Wu (2006)
  - Xue and Ekici (2013)
- Distributed Scheduling and Routing Policies
  - Tassiulas (1998)
  - Lee, Modiano, and Le (2009)
- Delay based Control
  - Cui, Lau, Wang, Huang, and Zhang (2012)
  - Singh and Kumar (2016)

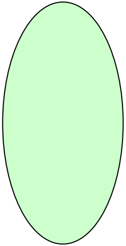
Control



Action Space Π

Control

$\pi^*(1) \to \pi^*(2) \to \pi^*(3) \to \dots$



Capacity Region Λ: Set of arrival rates for which a stabilizing policy exists.

Definition

Stability

$\lim_{T\to\infty} \sup \frac{1}{T} \sum_{\tau=0}^{T-1} \mathbb{E}[q_i^c(\tau)] < \infty$

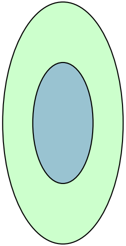
MaxWeight Policy

$p^*(t) = \arg_{p\in\mathcal{P}} \max \sum_{ij} \Delta_{ij} r_{ij}(p)$

- $\Delta_{ij} = \max_c (q_i^c - q_j^c)^+$
- Stabilizes all $\lambda \in \Lambda$

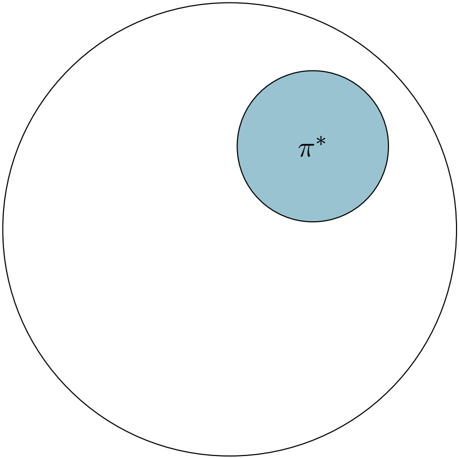
Control

$\pi(1) \to \pi(2) \to \pi(3) \to \dots$



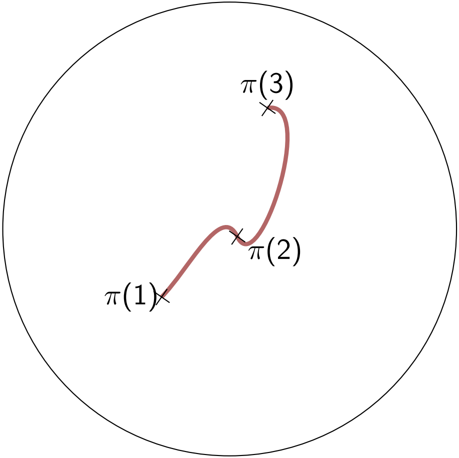
Capacity Region Λ

ϵ-Optimal Control



Action Space Π

Improvement over time



$$f(\pi(1)) \leq f(\pi(2)) \leq f(\pi(3))...$$

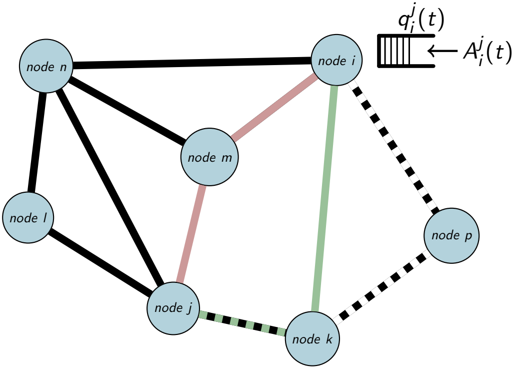
Providing QoS

- Priority based
- QoS vs Stability

Aim

- Distributed Algorithm
- Provide QoS
  - End to End Mean Delay Guarantee
  - End to End Hard Deadline Guarantee

End to End Delay

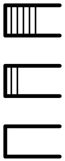


Multiple Paths between Source and Destination

Algorithm

Dynamic Priority at a Node

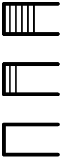
Higher Queue Length $\Longrightarrow$ Higher Priority



Algorithm

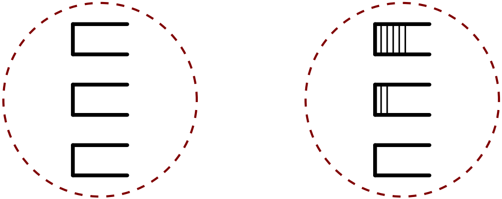
Dynamic Priority at a Node

QoS not satisfied $\Longrightarrow$ Higher Priority



Algorithm

Dynamic Priority across Nodes



Lower Queue Length	Higher Queue Length
Fewer QoS Packets	More QoS Packets
$\Longrightarrow$ Receiver	$\Longrightarrow$ Transmitter

Algorithm Q-Dep

- Virtual queue length $q_i = \sum_{c \in F} h^c(q_i^c)$

$$h^c(x) = \begin{cases} \theta x^2, & \text{if QoS of flow c not satisfied} \\ x, & \text{otherwise} \end{cases}$$

- Node becomes transmitter wp $\frac{u_i}{U^*}$; receiver wp $1 - \frac{u_i}{U^*}$
  - $u_i = \min(q_i, B)$; $U^* = \sum u_i$; Gossiping

Algorithm Q-Dep

Gossiping

- Node i has $Val(i)$
- Node i picks a neighbour wp $\frac{1}{N}$
- Update $Val(i), Val(j) \leftarrow \min(Val(i), Val(j))$
- If $Val(i) \sim \exp(Z_i)$, $\min(Val(i), Val(j)) \sim \exp(Z_i + Z_j)$

Algorithm Q-Dep

- Each transmitter i picks power $p_i \sim \mathcal{U}[0, p_{max}]$
- Over link ij schedule the flow

$$c_{ij}^* = \begin{cases} \arg_{c \in F} \max(h^c(q_i^c) - h^c(q_j^c))^+, & \text{wp } \sigma \\ \arg_{c \in F} \max(q_i^c - q_j^c)^+, & \text{otherwise} \end{cases}$$

- Compare current and previous slot, choose better

Stability

Theorem

The Algorithm Q-Dep stabilizes the network for any arrival rate vector $\lambda \in \rho\Lambda$ *where* $\rho < 1 - (\alpha_1 + (1-\alpha_1)\alpha_2) - 2\sqrt{\frac{\beta_2}{\beta_1}}$.[1]

- $\alpha_1 \in (0, 1)$
- $\alpha_3 \in \left(0, \frac{1}{2NB}\right)$
- $\beta_3 \in (0, 1)$
- $\beta_1 = (1-\beta_3)\left(\frac{\epsilon}{2(1-\alpha_3^2)N^{3.5}B^2}\right)^N$
- $\alpha_2, \beta \in (0, 1)$
- $\beta_2 = \beta + \sigma(1-\beta)$

[1] Ashok Krishnan.K.S, Vinod Sharma, NCC 2017.

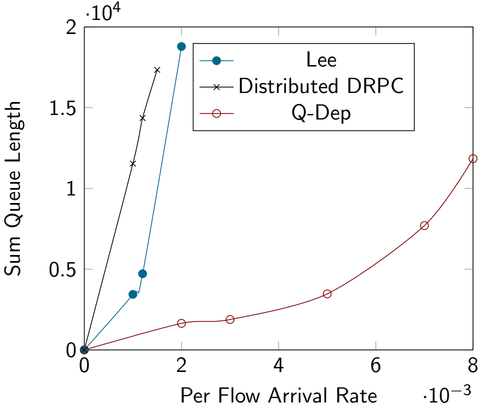
Stability

- A lower bound
- Stability region may be larger
- QoS vs Stability

Proof Sketch

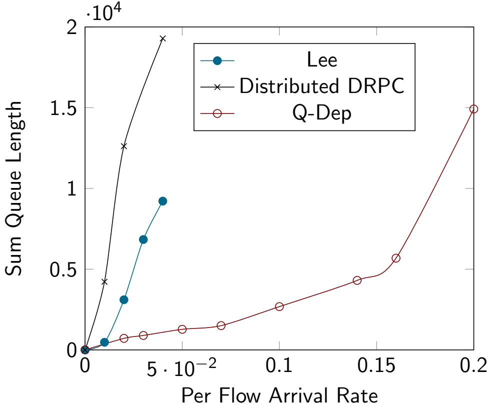
- $f(\pi(t)) \geq (1 - \psi_1) f(\pi^*(t))$ with positive probability
- $f(\pi(t+1)) \geq (1 - \psi_2) f(\pi(t))$ with positive probability

Performance



Stability Region for Different Algorithms for a Network of 20 Nodes and 5 Flows

Performance



Stability Region for Different Algorithms for a Network of 20 Nodes and 15 Flows

Performance

One flow with mean delay requirement
(Network 1: 10 nodes, 7 flows Network 2: 15 nodes, 10 flows)

Network 1		Network 2	
Delay Target (slots)	Delay Achieved (slots)	Delay Target (slots)	Delay Achieved (slots)
200	202	350	353
180	181	300	292
150	152	230	236
120	121	200	212
100	100	180	193
80	83	150	160
60	61	120	149

Performance

Two flows with mean delay requirement
(15 nodes, 7 flows)

Flow 1		Flow 2	
Delay Target (slots)	Delay Achieved (slots)	Delay Target (slots)	Delay Achieved (slots)
300	308	300	330
250	248	250	256
200	210	250	270
150	169	200	202
180	182	180	189
160	185	160	179

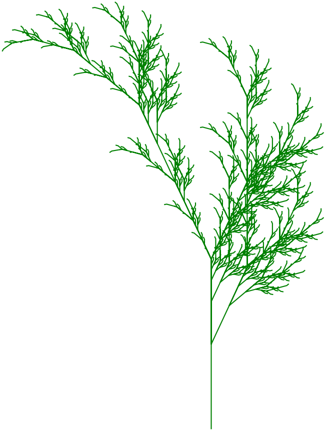
Performance

Two mean Delays and one hard deadline
(10 nodes, 8 flows)
Hard Deadline = 70 slots

Flow 1			Flow 2			Flow 3		
Delay Target (slots)	Delay Achieved (slots)	Mean Delay in Lee (slots)	Delay Target (slots)	Delay Achieved (slots)	Mean Delay in Lee (slots)	Drop ratio Target	Drop ratio Achieved	Drop Ratio in Lee
30	31		40	41		5%	5.1%	
30	31	127	40	41	104	3%	3%	52.7%
30	31		40	40		2%	2%	

Conclusion

- Distributed Algorithm
- Stability
- QoS provisions



Thank You.