Throughput-Optimal Broadcast in Wireless Networks with Point-to-Multipoint Transmissions

Abhishek Sinha

Laboratory for Information and Decision Systems

MIT

MobiHoc, 2017

April 18, 2017

Outline

Motivation

- We consider the problem of optimally broadcasting packets in a multi-hop wireless adhoc network

Motivation

- We consider the problem of optimally broadcasting packets in a multi-hop wireless adhoc network
- A primary measure of efficiency is *throughput-optimality*, i.e., policies that achieve the entire capacity region
- Vast literature for the Unicast problem (Backpressure policy), not so much for other flow problems
  - Packet duplications are harder to deal with (no flow conservations)

Introduction

- We study the Generalized Flow Problem and design throughput-optimal policies.
- A *Fundamental* problem with wide ranging applications: Internet routing, in-network function computations, live multi-media streaming, military communications etc.
- Topics of this talk:

  1. Broadcast: Specialized dynamic algorithms that solve the throughput-optimal broadcasting problem
     - It admits an inherently decentralized solution

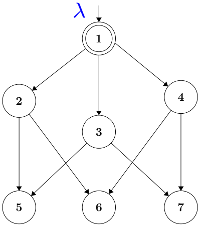
Introduction

- We study the Generalized Flow Problem and design throughput-optimal policies.
- A *Fundamental* problem with wide ranging applications: Internet routing, in-network function computations, live multi-media streaming, military communications etc.
- Topics of this talk:
  1. Broadcast: Specialized dynamic algorithms that solve the throughput-optimal broadcasting problem
     - It admits an inherently decentralized solution
  2. Generalized Flow: A general algorithmic paradigm that efficiently solves *all flow problems* (unicast+broadcast+multicast+anycast).

System Model

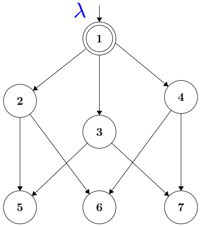
- The Wireless Network is represented by a graph $\mathcal{G}(V, E)$, where each node has an omnidirectional antenna.
- Packets arrive at a source node r i.i.d. at every slot at rate λ.
- Due to the *local broadcast* nature of the wireless medium, packets transmitted by a node i is heard at all of its out-neighbor $j \in \partial^+(i)$.
- As a result, if two or more in-neighbors of a node transmits at a slot, it results in a collision.
- This talk considers collision-free schedules only. The set of all feasible collision-free node activations is given by $\mathcal{M}$.

An Illustrative Example

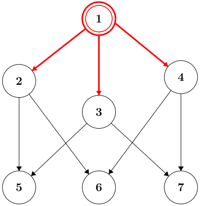


A wireless network

An Illustrative Example

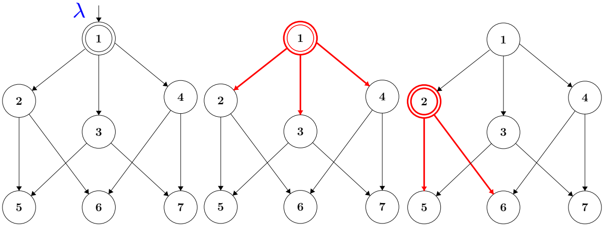


A wireless network



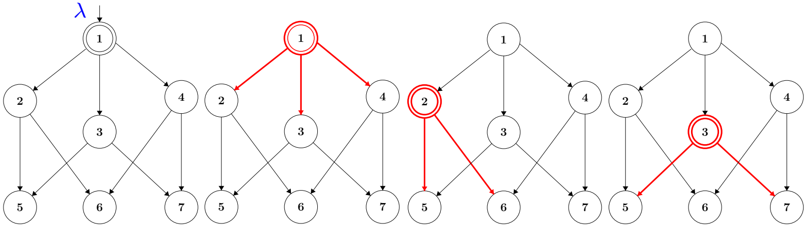
Activation $\boldsymbol{s}_1$

An Illustrative Example



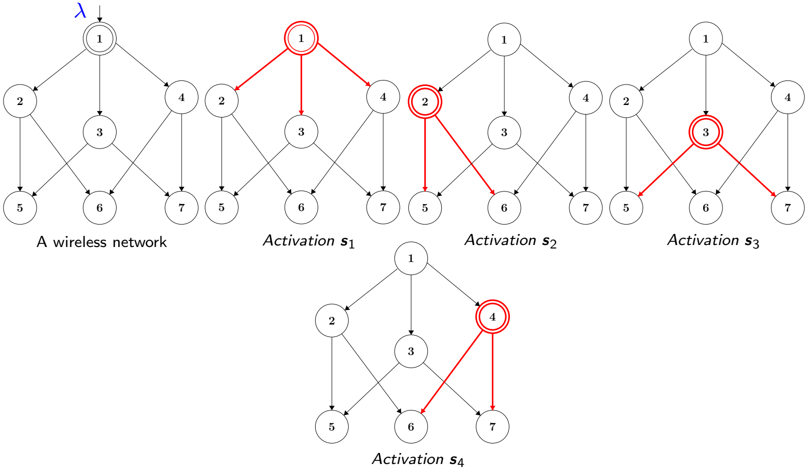
A wireless network | *Activation* s_1 | *Activation* s_2

An Illustrative Example



A wireless network | *Activation* $\mathbf{s}_1$ | *Activation* $\mathbf{s}_2$ | *Activation* $\mathbf{s}_3$

An Illustrative Example



A wireless network and its feasible link activations under the primary interference constraints.
$\mathcal{M} = \{s_1, s_2, s_3, s_4\}$

Wireless Broadcast: Problem Formulation

A feasible broadcast policy $\pi \in \Pi$ executes the following two actions at every slot t :

- Node Activation $\pi(\mathcal{A})$: Activates a subset of nodes $\boldsymbol{s}(t) \in \mathcal{M}$ subject to the underlying interference constraints.
- **Packet Scheduling $\pi(\mathcal{S})$:** The activated nodes *locally broadcasts* a set of packets subject to the capacity/power constraints of the nodes.

Wireless Broadcast: Problem Formulation

A feasible broadcast policy $\pi \in \Pi$ executes the following two actions at every slot t :

- Node Activation $\pi(\mathcal{A})$: Activates a subset of nodes $\boldsymbol{s}(t) \in \mathcal{M}$ subject to the underlying interference constraints.
- **Packet Scheduling $\pi(\mathcal{S})$:** The activated nodes *locally broadcasts* a set of packets subject to the capacity/power constraints of the nodes.

- Let $R^\pi(T)$ denote the number of packets received in common by all nodes under the action of a broadcast policy π.
- The objective is to design a policy π such that for all $\lambda < \lambda^*$

$$\liminf_{T\to\infty} \frac{R^\pi(T)}{T} = \lambda, \quad \text{w.p. } 1,$$

where λ^* is the broadcast capacity of the network.

Outline

Hardness Result

Our first result in this point-to-multipoint broadcast setting is the following:

Theorem

WIRELESS BROADCAST is NP-complete.

Hardness Result

Our first result in this point-to-multipoint broadcast setting is the following:

Theorem

WIRELESS BROADCAST is NP-complete.

- We also show that the problem remains hard even with the additional restriction of DAG topology and no activation constraints.

Hardness Result

Our first result in this point-to-multipoint broadcast setting is the following:

Theorem

WIRELESS BROADCAST is NP-complete.

- We also show that the problem remains hard even with the additional restriction of DAG topology and no activation constraints.
- This is surprising, because we showed earlier [Sinha et al, 2015, 2016] that the problem is efficiently solvable in case of wireless DAGs with point-to-point links.

Hardness Result

Our first result in this point-to-multipoint broadcast setting is the following:

Theorem

Wireless Broadcast is NP-complete.

- We also show that the problem remains hard even with the additional restriction of DAG topology and no activation constraints.
- This is surprising, because we showed earlier [Sinha et al, 2015, 2016] that the problem is efficiently solvable in case of wireless DAGs with point-to-point links.
- The hardness comes from the requirement of optimally distributing the packets, which is intimately related to Boolean Constraint Satisfaction, described next.

Hardness Reduction: Proof Sketch

Proof: MONOTONE NOT ALL EQUAL 3-SAT (MNAE 3-SAT) $\implies$ WIRELESS BROADCAST.

Hardness Reduction: Proof Sketch

Proof: MONOTONE NOT ALL EQUAL 3-SAT (MNAE 3-SAT) $\implies$ WIRELESS BROADCAST.

MNAE-3SAT: Given a CNF formula $C = \wedge_i (x_{i_1} \vee x_{i_2} \vee x_{i_3})$ with no complemented variable.

Problem: Does there exist a satisfying assignment such that each clause contain at least one false literal? (Y/N)

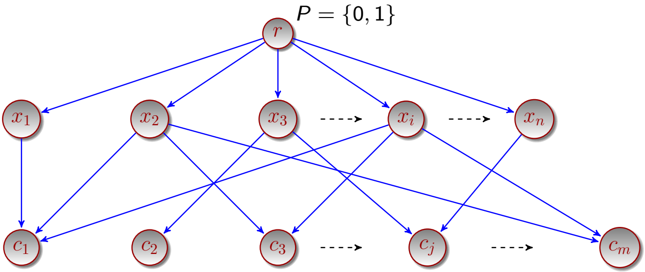
Hardness Reduction: Proof Sketch

Proof: MONOTONE NOT ALL EQUAL 3-SAT (MNAE 3-SAT) $\implies$ WIRELESS BROADCAST.

MNAE-3SAT: Given a CNF formula $C = \wedge_i (x_{i_1} \vee x_{i_2} \vee x_{i_3})$ with no complemented variable.

Problem: Does there exist a satisfying assignment such that each clause contain at least one false literal? (Y/N)

Question: Can we broadcast two packets from the source r in two slots?



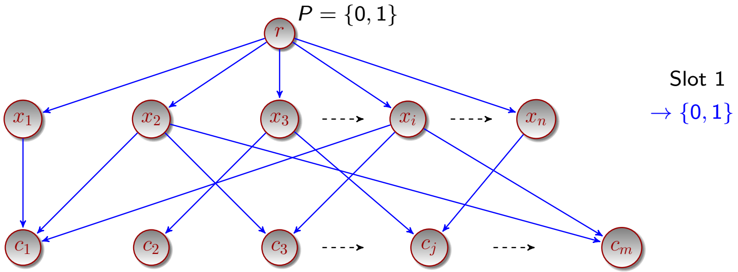
Hardness Reduction: Proof Sketch

Proof: MONOTONE NOT ALL EQUAL 3-SAT (MNAE 3-SAT) $\implies$ WIRELESS BROADCAST.

MNAE-3SAT: Given a CNF formula $C = \wedge_i (x_{i_1} \vee x_{i_2} \vee x_{i_3})$ with no complemented variable.

Problem: Does there exist a satisfying assignment such that each clause contain at least one false literal? (Y/N)

Question: Can we broadcast two packets from the source r in two slots?



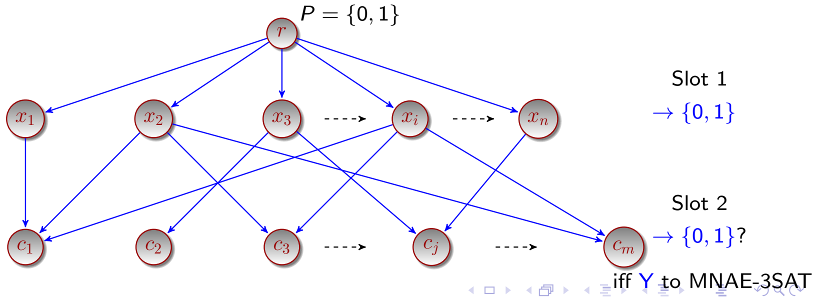
Hardness Reduction: Proof Sketch

Proof: MONOTONE NOT ALL EQUAL 3-SAT (MNAE 3-SAT) $\implies$ WIRELESS BROADCAST.

MNAE-3SAT: Given a CNF formula $C = \wedge_i (x_{i_1} \vee x_{i_2} \vee x_{i_3})$ with no complemented variable.

Problem: Does there exist a satisfying assignment such that each clause contain at least one false literal? (Y/N)

Question: Can we broadcast two packets from the source r in two slots?



Outline

1. Introduction
2. Hardness of Wireless Broadcasting
3. Throughput-Optimal Broadcast Policy
4. Numerical Results
5. Conclusion

Routing of Packets: Connected Dominating Sets (CDS)

CDS: Defintion

A connected dominating set in a directed graph $\mathcal{G}(V, E)$ and root r is a set of vertices $S \subseteq V$ such that:

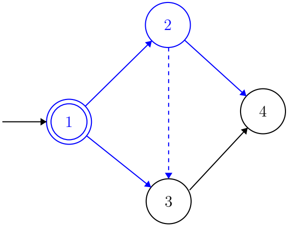
- For every $v \in V$, $\exists$ a directed path $\pi = r \to v_1 \to v_2 \ldots v_i \to v$, where all nodes, excepting possibly v, are in the set S.

Routing of Packets: Connected Dominating Sets (CDS)

CDS: Defintion

A connected dominating set in a directed graph $\mathcal{G}(V, E)$ and root r is a set of vertices $S \subseteq V$ such that:

- For every $v \in V$, $\exists$ a directed path $\pi = r \to v_1 \to v_2 \ldots v_i \to v$, where all nodes, excepting possibly v, are in the set S.



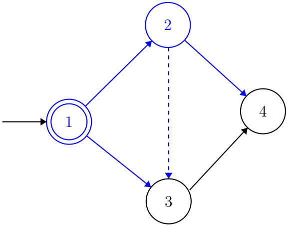
The sets $S_1 = \{1, 2\}, S_2 = \{1, 3\}$ are two CDS in this network.

Routing of Packets: Connected Dominating Sets (CDS)

CDS: Defintion

A connected dominating set in a directed graph $\mathcal{G}(V, E)$ and root r is a set of vertices $S \subseteq V$ such that:

- For every $v \in V$, $\exists$ a directed path $\pi = r \to v_1 \to v_2 \dots v_i \to v$, where all nodes, excepting possibly v, are in the set S.



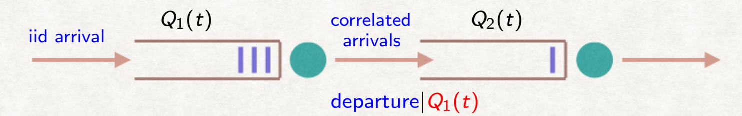
The sets $S_1 = \{1, 2\}, S_2 = \{1, 3\}$ are two CDS in this network.

Key observation: Route of a Packet

Every packet must be transmitted sequentially by a CDS in order to be broadcasted.

Design of UMW: Motivation and Insight

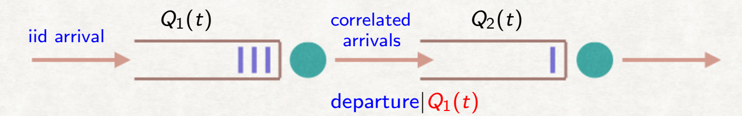
- **Observation:** Because of coupling, networked queues are harder to analyze and control.



IID arrivals to Q_1 causes correlated arrivals to Q_2

Design of UMW: Motivation and Insight

- **Observation:** Because of coupling, networked queues are harder to analyze and control.



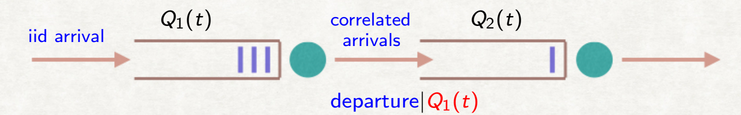
IID arrivals to Q_1 causes correlated arrivals to Q_2

- This motivates us to obtain a relaxed system, easier to analyze, yet, preserves properties of interest (e.g., stability).

Question: How to obtain a good relaxation? Which constraints to relax?

Design of UMW: Motivation and Insight

- **Observation:** Because of coupling, networked queues are harder to analyze and control.



IID arrivals to Q_1 causes correlated arrivals to Q_2

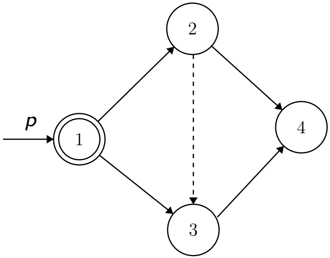
- This motivates us to obtain a relaxed system, easier to analyze, yet, preserves properties of interest (e.g., stability).

Question: How to obtain a good relaxation? Which constraints to relax?

Ans: The Precedence Constraints!

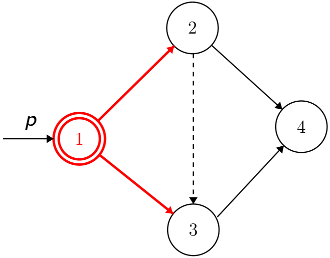
Precedence Constraint: Example

Consider an incoming packet p with the specified broadcasting route $CDS_p = \{1, 2\}$.



Precedence Constraint: Example

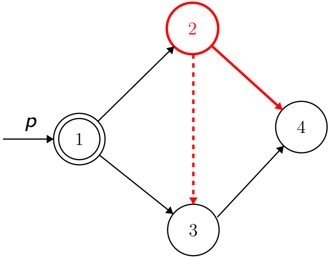
Consider an incoming packet p with the specified broadcasting route $CDS_p = \{1, 2\}$.



Slot 1

Precedence Constraint: Example

Consider an incoming packet p with the specified broadcasting route $CDS_p = \{1, 2\}$.

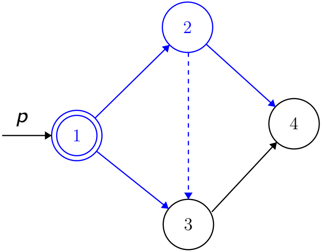


Slot 2

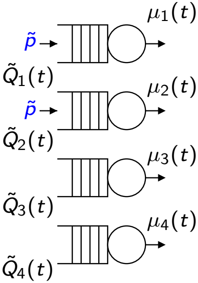
Observation: Due to the precedence, the packet p is transmitted by Node 2 after it has been transmitted by Node 1

Precedence Relaxation: Example

We maintain a virtual system where the packet p is injected to the virtual queues $\tilde{Q}_1, \tilde{Q}_2$ immediately upon arrival.



A Wireless Network $\mathcal{G}$



Virtual Queues

Virtual Queues: Operation

Formally,

1. Associate a virtual queue $\tilde{Q}_v(t)$ with each node v of the graph.
2. Upon packet arrival:
   - Determine a *CDS* $T_p^*(t)$ for the packet p
   - Immediately inject a new virtual packet to each virtual queue along in the CDS
     - This amounts to incrementing the queue counters in the CDS

Virtual Queues: Operation

Formally,

1. Associate a virtual queue $\tilde{Q}_v(t)$ with each node v of the graph.
2. Upon packet arrival:
   - Determine a *CDS* $T^*_p(t)$ for the packet p
   - Immediately inject a new virtual packet to each virtual queue along in the CDS
     - This amounts to incrementing the queue counters in the CDS
3. Serve the virtual packets at the rate $\boldsymbol{\mu}^*(t)$ as long as the corresponding virtual queues are non-empty
   - Subject to the same link scheduling constraints ($\boldsymbol{\mu}^*(t) \in \mathcal{M}$)
   - Don't care whether the physical queue is empty or not.

Virtual Queues: Operation

Formally,

1. Associate a virtual queue $\tilde{Q}_v(t)$ with each node v of the graph.
2. Upon packet arrival:
   - Determine a *CDS* $T_p^*(t)$ for the packet p
   - Immediately inject a new virtual packet to each virtual queue along in the CDS
     - This amounts to incrementing the queue counters in the CDS
3. Serve the virtual packets at the rate $\boldsymbol{\mu}^*(t)$ as long as the corresponding virtual queues are non-empty
   - Subject to the same link scheduling constraints ($\boldsymbol{\mu}^*(t) \in \mathcal{M}$)
   - Don't care whether the physical queue is empty or not.

Question: How to design the optimal controls: $T_p^*(t)$ and $\mu^*(t)$?

Dynamics of the Virtual Queues $\tilde{\boldsymbol{Q}}(t)$

The virtual queue lengths can be mathematically identified with an n-dimensional vector taking values in $\mathbb{Z}_+^n$.

Dynamics of the Virtual Queues $\tilde{\boldsymbol{Q}}(t)$

The virtual queue lengths can be mathematically identified with an n-dimensional vector taking values in $\mathbb{Z}_+^n$.

- Denote the (controlled) arrival to the VQ $\tilde{Q}_i$ by $\tilde{A}_i(t)$. Then, the virtual queues evolve as:

$$\tilde{Q}_i(t+1) = \left(\tilde{Q}_i(t) + \tilde{A}_i(t) - \mu_i(t)\right)^+, \quad \text{(Lindley recursion)} \tag{1}$$

- Note that, the arrivals to the virtual queues $\left(\tilde{A}_i(t), i \in V\right)$ are explicit control variables at the source.
- Unlike the original system, given the controls, the virtual queues are independent of each other. This makes their exact analysis tractable.

Stabilizing Controls for $\tilde{Q}(t)$: Drift Analysis

- A natural first-step is to design $\pi^{\mathtt{UMW}} \equiv \big(\boldsymbol{A}(t), \boldsymbol{\mu}(t)\big)_{t\geq 0}$, such that, it stabilizes the virtual system $\{\tilde{\boldsymbol{Q}}(t)\}_{t\geq 0}$.
- The policy consists of the routing decisions : routing $\boldsymbol{A}^{\pi}(t)$, and scheduling $\boldsymbol{\mu}^{\pi}(t)$.
- Intuition: This control is *likely to stabilize* the physical queues as well
  - However, note that the dynamics of the physical queues depend explicitly on the packet scheduling policy (e.g., FIFO, LIFO etc.). We will come to this issue later.

Stabilizing Controls for $\tilde{\boldsymbol{Q}}(t)$: Drift Analysis

- A natural first-step is to design $\pi^{\mathtt{UMW}} \equiv \big(\boldsymbol{A}(t), \boldsymbol{\mu}(t)\big)_{t\geq 0}$, such that, it stabilizes the virtual system $\{\tilde{\boldsymbol{Q}}(t)\}_{t\geq 0}$.
- The policy consists of the routing decisions : routing $\boldsymbol{A}^{\pi}(t)$, and scheduling $\boldsymbol{\mu}^{\pi}(t)$.
- Intuition: This control is *likely to stabilize* the physical queues as well
  - However, note that the dynamics of the physical queues depend explicitly on the packet scheduling policy (e.g., FIFO, LIFO etc.). We will come to this issue later.
- To stabilize the virtual queues, we choose the control that minimizes the drift of the Quadratic Lyapunov (potential) function of the Virtual Queues.

Derivation of the Control-Policy

- Define a Quadratic Lyapunov (potential) function

$$L(\tilde{\boldsymbol{Q}}(t)) \stackrel{\text{def}}{=} \sum_{i \in V} \tilde{Q}_i^2(t)$$

- The one-slot drift of $L(\tilde{\boldsymbol{Q}}(t))$ under any admissible policy π may be computed to be

$$\begin{aligned} \Delta^{\pi}(t) &\stackrel{\text{def}}{=} L(\tilde{\boldsymbol{Q}}(t+1)) - L(\tilde{\boldsymbol{Q}}(t)) \\ &\leq B + 2\Big(\underbrace{\sum_{i \in V} \tilde{Q}_i(t) A(t) \mathbb{1}(i \in T^{\pi}(t))}_{(a)} - \underbrace{\sum_{i \in V} \tilde{Q}_i(t) \mu_i^{\pi}(t)}_{(b)} \Big) \end{aligned} \quad (2)$$

Where $T^{\pi}(t) \in \mathcal{T}$ and $\boldsymbol{\mu}^{\pi}(t) \in \mathcal{M}$ are routing and activation control variables chosen for slot t.

- The drift upper-bound (2) has a nice separable form and may be minimized over the routing and activation controls individually.

Optimal Routing Policy $T_p^*(t)$

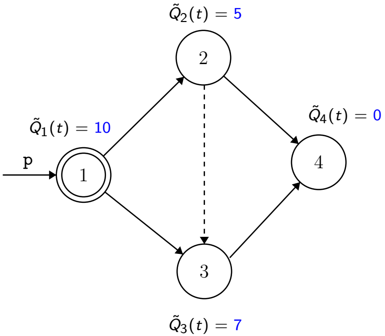
Let $\mathcal{T}$ denote the set of all CDS in $\mathcal{G}$. Minimizing the routing term (a), we get the following optimal routing policy.

Optimal Routing : $\boldsymbol{T_p^*(t)}$

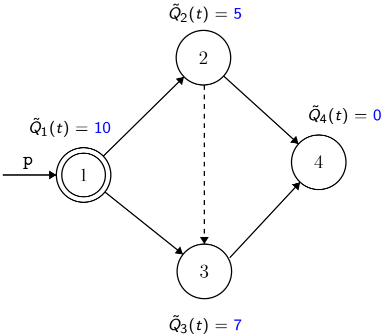
$$T_p^*(t) \in \arg\min_{T \in \mathcal{T}} \sum_{i \in V} \tilde{Q}_i(t) \mathbb{1}(i \in T)$$

In other words, the drift minimizing routing policy is to route the incoming packet along the Minimum Weight CDS, where each node i is weighted by the corresponding virtual queue $\tilde{Q}_i(t)$.

Example of Optimal Routing



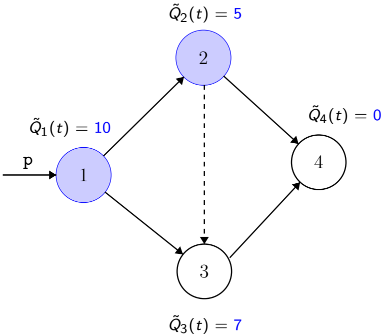
Example of Optimal Routing



Weight of CDS $\{1, 2\} = 15$

Weight of CDS $\{1, 3\} = 17$

Example of Optimal Routing



Weight of CDS $\{1, 2\} = 15$

Weight of CDS $\{1, 3\} = 17$

Chosen route=MCDS= $\{1, 2\}$

Optimal Node Scheduling Policy $\mu^*(t)$

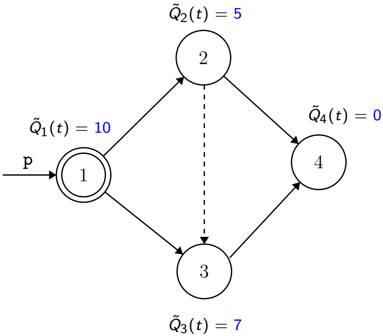
Let $\mathcal{M}$ denote the set of all non-interfering activations in $\mathcal{G}$. Minimizing the scheduling term (b) in the drift expression, we get the following optimal scheduling policy.

Optimal Scheduling : $\boldsymbol{\mu}^*(t)$

$$\mu^*(t) \in \arg\max_{M \in \mathcal{M}} \sum_{i \in V} \tilde{Q}_i(t) \mathbb{1}(i \in M)$$

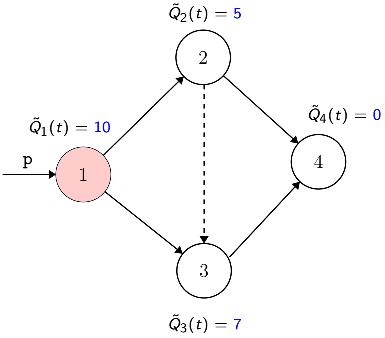
In other words, the drift minimizing node scheduling policy is to schedule the Max-Weight activation, where each node i is weighted by the corresponding virtual queue length $\tilde{Q}_i(t)$.

Example of Optimal Scheduling



Due to interference,
can activate only one node per slot

Example of Optimal Scheduling



Due to interference, can activate only one node per slot

Optimal Schedule = Activate the Node 1

Stability of the Virtual Queue

Theorem 6: Strong Stability of $\tilde{\boldsymbol{Q}}(t)$

Under the above routing and scheduling policy, for all arrival rate $\boldsymbol{\lambda} \leq \lambda^*$ the virtual queue process is Strongly stable and has a limiting M.G.F, i.e.,

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T}\sum_{i}\mathbb{E}(\tilde{Q}_i(t)) \leq B$$

and,

$$\limsup_{T\to\infty} \mathbb{E}(\exp(\theta^* \sum_{i} \tilde{Q}_i(t))) \leq C$$

for some finite B, C and strictly positive θ^*.

Stability of the Virtual Queue

Theorem 6: Strong Stability of $\tilde{\boldsymbol{Q}}(t)$

Under the above routing and scheduling policy, for all arrival rate $\boldsymbol{\lambda} \leq \lambda^*$ the virtual queue process is Strongly stable and has a limiting M.G.F, i.e.,

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_i \mathbb{E}(\tilde{Q}_i(t)) \leq B$$

and,

$$\limsup_{T\to\infty} \mathbb{E}(\exp(\theta^* \sum_i \tilde{Q}_i(t))) \leq C$$

for some finite B, C and strictly positive θ^*.

The above leads to the following :

Lemma: Sample Path bound on Virtual Queues

Under the same condition, we have

$$\sum_i \tilde{Q}_i(t) = \mathcal{O}(\log t), \quad \text{a.s.}$$

Optimal Control of the Physical Queues : Packet Scheduling

- How do we decide which packet to transmit over a link at any given time slot?
  - Why does it matter? Cannot we just use FCFS?
- Nearest to Origin (NTO) policy [Gamarnik, 1998]
- Extended Nearest to Origin policy (ENTO): When multiple packets contend for an edge, schedule the one which has traversed the least number of edges
  - Extension of NTO to general flow problems

Optimal Control of the Physical Queues : Packet Scheduling

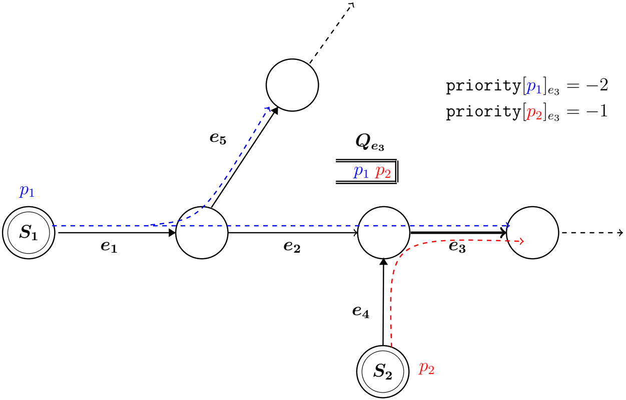
- How do we decide which packet to transmit over a link at any given time slot?
  - Why does it matter? Cannot we just use FCFS?
- Nearest to Origin (NTO) policy [Gamarnik, 1998]
- Extended Nearest to Origin policy (ENTO): When multiple packets contend for an edge, schedule the one which has traversed the least number of edges
  - Extension of NTO to general flow problems

Theorem 7: Stability of the Physical Queues

The overall UMW policy is throughput-optimal.

Proof uses the previous almost sure arrival bound on a typical sample path with an inductive argument on the edges.

ENTO: Example



Packet p_1 has higher priority than p_2 to cross e_3 as it has traversed less number of edges

Proof Ideas for Theorem 7: Stability of the Physical Queues

1. Observation: Since routes are fixed at source, total number of arrival $\tilde{A}_e(t_1, t_2)$ in interval $[t_1, t_2]$ at virtual queue $\tilde{Q}_i = A_i(t_1, t_2)$ total number of packets that wish to be trabsmitted by the node i in the physical network sometime in future.

Proof Ideas for Theorem 7: Stability of the Physical Queues

1. Observation: Since routes are fixed at source, total number of arrival $\tilde{A}_e(t_1, t_2)$ in interval $[t_1, t_2]$ at virtual queue $\tilde{Q}_i = A_i(t_1, t_2)$ total number of packets that wish to be trabsmitted by the node i in the physical network sometime in future.

2. Theorem 6 (Stability of the Virtual Queues) + Skorokhod Map representation + Almost Sure Bound $\implies$

$$A_i(t_0, t) \leq S_i(t_0, t) + \mathcal{O}(\log(t)), \ \forall i \in V, \quad t_0 \leq t, \quad \text{w.p. } 1$$

- This essentially implies that *none* of the nodes are overloaded under the UMW scheduling policy

Proof Ideas for Theorem 7: Stability of the Physical Queues

1. Observation: Since routes are fixed at source, total number of arrival $\tilde{A}_e(t_1, t_2)$ in interval $[t_1, t_2]$ at virtual queue $\tilde{Q}_i = A_i(t_1, t_2)$ total number of packets that wish to be trabsmitted by the node i in the physical network sometime in future.

2. Theorem 6 (Stability of the Virtual Queues) + Skorokhod Map representation + Almost Sure Bound $\implies$

$$A_i(t_0, t) \leq S_i(t_0, t) + \mathcal{O}(\log(t)), \; \forall i \in V, \quad t_0 \leq t, \quad \text{w.p. } 1$$

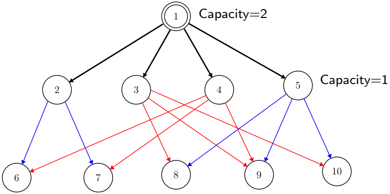
   - This essentially implies that *none* of the nodes are overloaded under the UMW scheduling policy

3. With the universal stability property of the ENTO packet scheduling policy it is finally shown that the physical queues are rate stable.

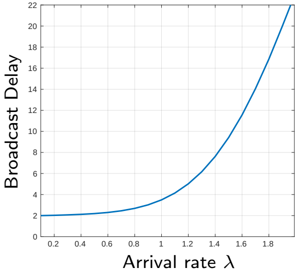
   - Involves induction on the number of hops from the source.

Outline

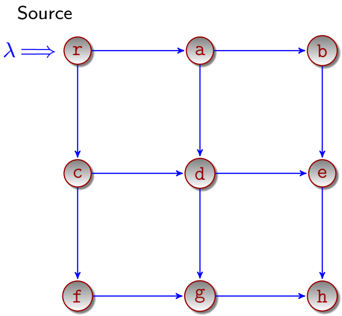
Broadcasting: Network without Interference



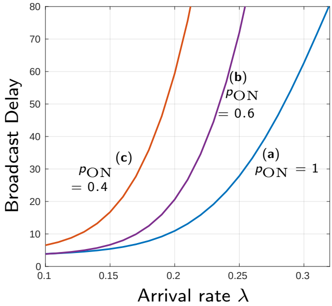
A wireless network with non-interfering channels. The broadcast capacity of the network is $\lambda^* = 2$.



Broadcasting Simulation: Time-Varying Wireless Network



The Grid Network



Plot of the broadcast delay incurred by the UMW policy as a function of the arrival rate λ in the 3×3 wireless grid network.

Outline

Conclusion

- Our understanding of network control theory has progressed enormously over the past 25 years, starting with the seminal Backpressure policy of Tassiulas and Ephremides (1992).
- We have derived a throughput-optimal algorithm, UMW, for broadcasting in wireless networks with **point-to-multipoint** links.
- This important problem was proposed by Massoulie and Twigg, and has remained open for last ten years.
- The virtual network framework used to solve the problem is surprisingly general and may be applied to other open problems in this area (e.g., Sinha, Modiano, INFOCOM '17).
- Opens up exciting new directions for research with lots of interesting problems.