Optimal Distributed Broadcast Algorithms for Wireless DAGs

Abhishek Sinha, Georgios Paschos, Chih-ping Li and Prof. Eytan Modiano,

Laboratory for Information and Decision Systems

Massachusetts Institute of Technology

Authors

Abhishek Sinha

Laboratory for Information and Decision Systems

Distributed Load Management in Anycast-based CDNs.

Optimal sequential wireless relay placement on a random lattice path.

Georgios Paschos

Principal Research Engineer at the Mathematical and Algorithmic Sciences Laboratory of Huawei Technologies, in Paris, since Nov. 2014.

Chih-ping Li

Qualcomm.

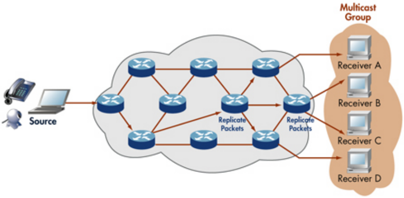
Prof. Eytan Modiano

Principal Investigator of Communications and Networking Research Group, LIDS, MIT.

The Network Broadcast Problem (one-to-all)

Need to disseminate messages, generated at a source, to all other nodes in a network.

In wired settings, applications in live video streaming, software updates, synchronization of distributed servers etc.



A wired network

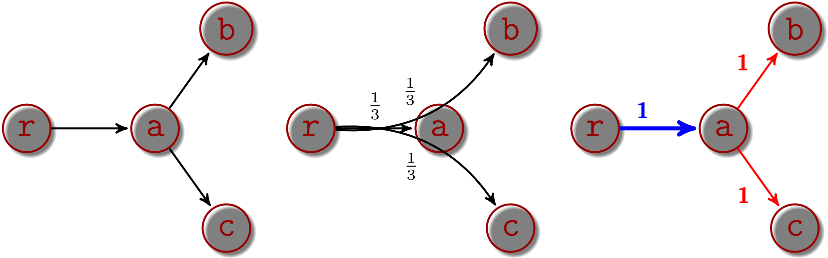
A wireless adhoc network

In wireless settings, applications in military communications, disaster management over a mobile adhoc network (MANET).

Naïve approach and its limitation

Flooding? **Broadcast Storm**.

Establish independent point-to-point unicast connections from source to all *other* nodes.

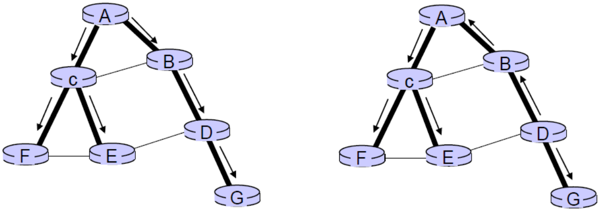


Throughput limitation: redundant transmissions in the link $\mathtt{r} \longrightarrow \mathtt{a}$. ✗

Question How to route/split the packets in an intelligent way ?

Spanning Tree Broadcast

Broadcast through a spanning tree :Acyclic.
Avoid redundant packet transmission.

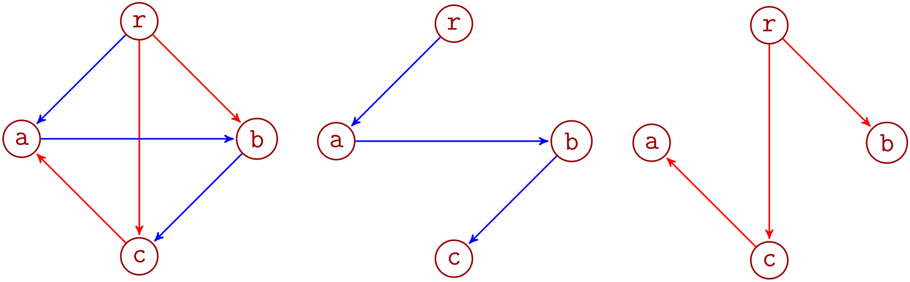


(a) broadcast initiated at A (b) broadcast initiated at D

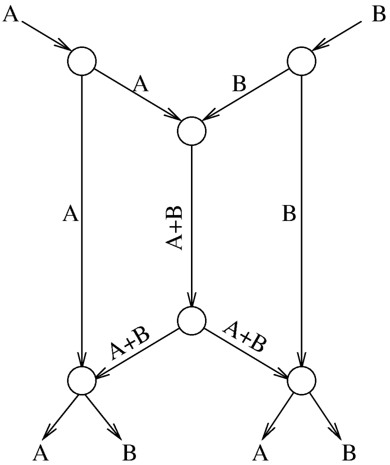
Limitation of Spanning Tree

To compute and rebuild a spanning tree in a **frequently changed network**.

Efficient bandwidth use?

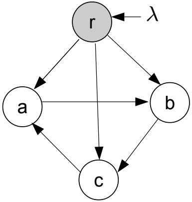


Network Coding



Interference Model

Packet transmissions are subject to wireless interference constraints.

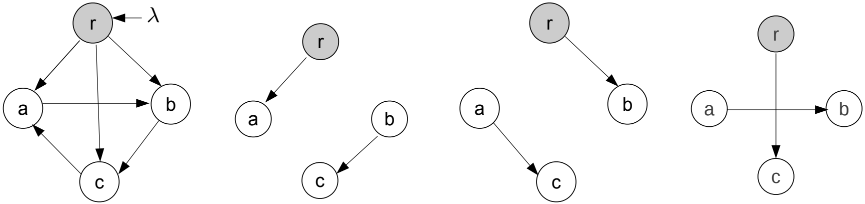


A wireless network

Interference Model

Packet transmissions are subject to wireless interference constraints.

Primary Interference Model



A wireless network — Activation s_1 — Activation s_2 — Activation s_3

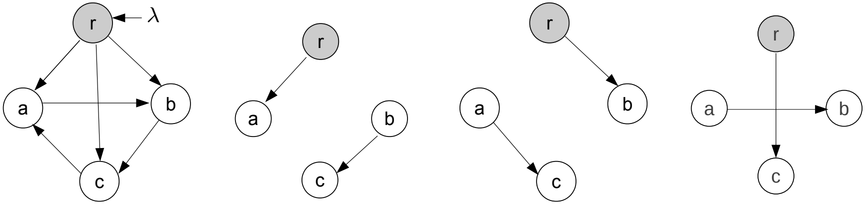
System Model

The network-topology is represented as a directed graph $\mathcal{G}(V, E, c, \mathcal{S})$.

$\boldsymbol{s} = (s_e, e \in E) \in \mathcal{S}$: a binary vector.
$s_e = 1$: link e is activated at mode s.

Time is slotted.

Source $r \in V$ generates packets at rate $\lambda > 0$.



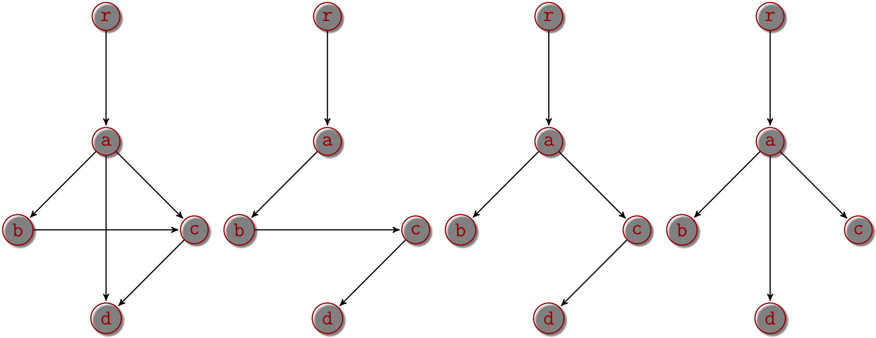
A wireless network   Activation $\boldsymbol{s}_1$ Activation $\boldsymbol{s}_2$ Activation $\boldsymbol{s}_3$

Nodes can duplicate and relay packets to their neighbors.

Challenge: Traditional queuing theory does not apply due to packet duplications.

Previous Works

Pre-computes the set of all spanning trees offline in wireline networks [Sarkar and Tassiulas, 2005].



- Impractical for large networks and time-varying wireless networks. ✗

Wireless case is studied with a fixed activation schedule [Towsley et al. 2008].

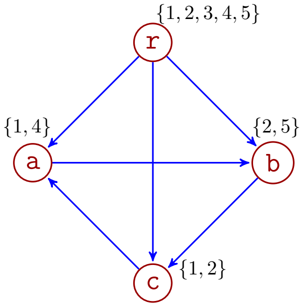
Contribution: To design an online algorithm to achieve the capacity in a DAG without computing the trees.

Feasible Policy Space Π

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

Link Activation $S^{\Pi}(t)$: Activate a subset of links (e.g., a matching) subject to the underlying interference constraints.
Packet Scheduling $A^{\Pi}(t)$: Transmit packets over the set of activated links.

An arbitrary $A^{\Pi}(t)$ is hard to describe : exponential growth with packets!

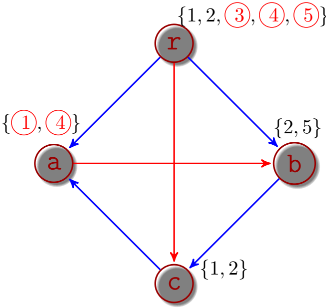


Feasible Policy Space Π

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

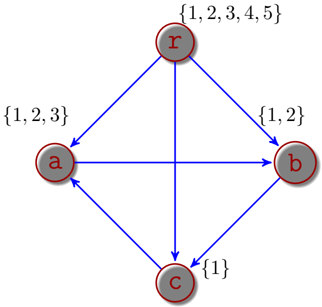
Link Activation $S^{\pi}(t)$: Activate a subset of links (e.g., a matching) subject to the underlying interference constraints.
Packet Scheduling $A^{\pi}(t)$: Transmit packets over the set of activated links.

An arbitrary $A^{\pi}(t)$ is hard to describe : exponential growth with packets!



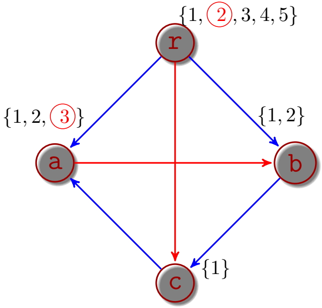
Policy sub-space $\Pi^{\text{in-order}}$

To simplify $A^{\Pi}(t)$, we consider the sub-space $\Pi^{\text{in-order}} \subset \Pi$ in which all packets are delivered at every node *in-order*.



Policy sub-space $\Pi^{\text{in-order}}$

To simplify $\boldsymbol{A^{\Pi}(t)}$, we consider the sub-space $\Pi^{\text{in-order}} \subset \Pi$ in which all packets are delivered at every node *in-order*.



Thus the system-state is represented by the vector

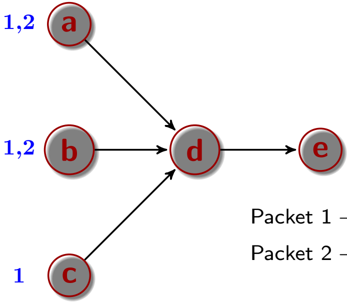
$$\boldsymbol{A}(t) = \{R_1(t), R_2(t), \ldots, R_{|V|}(t)\}$$

Where $R_i(t)$ is the *total number* of packets received by node i.

Policy-space $\Pi^* \subset \Pi^{\text{in-order}}$

$\Pi^* \subset \Pi^{\text{in-order}} \subset \Pi$

For all $\pi \in \Pi^*$, a packet p is eligible for transmission to node n iff all in-neighbors of node n contains the packet p.

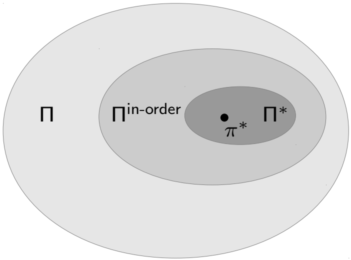


Packet 1 $\rightarrow$ node d ✓

Packet 2 $\rightarrow$ node d ✗

Policy hierarchies

All policies are un-restricted for link-activations $\boldsymbol{S^{\Pi}(t)}$



Π: all policies that perform arbitrary packet-forwarding

$\Pi^{\text{in-order}}$: policies that enforce in-order packet-forwarding

Π^*: policies that allow reception only if all in-neighbors have received the specific packet

Punchline : Π^* is optimal for DAG!

The Broadcast Capacity λ^*

for any $\lambda \leq \lambda^*$, every node other than source r can receive packets at rate λ

Observation : From source $\mathtt{r}$ to each node $\mathtt{t} \neq \mathtt{r}$,

$$\lambda^* \leq \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Thus,

$$\lambda^* \leq \min_{t \in V \setminus \{\mathtt{r}\}} \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Max-flow min-cut theorem

$$\lambda^* \leq \min_{U:\text{ a proper cut}} \sum_{e \in E_U} c_e, \quad E_U = \{(i,j) \in E \mid i \in U, j \notin U\}$$

for time varying case?

The Broadcast Capacity λ^*

for any $\lambda \leq \lambda^*$, every node other than source r can receive packets at rate λ

Observation : From source $\mathtt{r}$ to each node $\mathtt{t} \neq \mathtt{r}$,

$$\lambda^* \leq \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Thus,

$$\lambda^* \leq \min_{t \in V \setminus \{\mathtt{r}\}} \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Max-flow min-cut theorem

$$\lambda^* \leq \min_{U:\ \text{a proper cut}} \sum_{e \in E_U} c_e, \quad E_U = \{(i,j) \in E \mid i \in U, j \notin U\}$$

for time varying case?

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1} f(t)$$

The Broadcast Capacity λ^*

$$\boldsymbol{\beta}^\pi = (\beta_e^\pi, e \in E) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1} \boldsymbol{s}^\pi(t)$$

The Broadcast Capacity λ^*

$$\boldsymbol{\beta}^\pi = (\beta_e^\pi, e \in E) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1} \boldsymbol{s}^\pi(t)$$

$$\lambda^\pi \le \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi, \;\; E_U = \{(i,j) \in E \mid i \in U, j \notin U\}$$

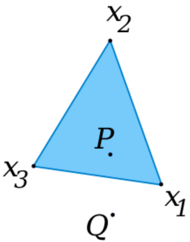
The Broadcast Capacity λ^*

$$\boldsymbol{\beta}^\pi = (\beta_e^\pi, e \in E) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1} \boldsymbol{s}^\pi(t)$$

$$\lambda^\pi \leq \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi, \quad E_U = \{(i,j) \in E \mid i \in U, j \notin U\}$$

$$\lambda^\pi \leq \sup_{\pi\in\Pi} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi \leq \max_{\boldsymbol{\beta}\in\text{conv}(\mathcal{S})} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e$$

Convex Combination



$\sum a_i s_i$, $a_i \geq 0$ and $\sum a_i = 1$

The Broadcast Capacity λ^*

$$\boldsymbol{\beta}^\pi = (\beta_e^\pi, e \in E) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1} \boldsymbol{s}^\pi(t)$$

$$\lambda^\pi \leq \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi, \;\; E_U = \{(i,j) \in E \mid i \in U, j \notin U\}$$

$$\lambda^\pi \leq \sup_{\pi\in\Pi} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi$$

$$\lambda^* \leq \max_{\boldsymbol{\beta}\in\text{conv}(\mathcal{S})} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e$$

The Broadcast Capacity λ^*

$$\boldsymbol{\beta}^\pi = (\beta_e^\pi, e \in E) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1} \boldsymbol{s}^\pi(t)$$

$$\lambda^\pi \leq \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi, \quad E_U = \{(i,j)\in E \mid i \in U, j \notin U\}$$

$$\lambda^\pi \leq \sup_{\pi\in\Pi} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e^\pi$$

$$\lambda^* \leq \max_{\boldsymbol{\beta}\in\text{conv}(\mathcal{S})} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\, \beta_e$$

Proved: In a DAG, exist a $\pi\in\Pi^*$ that λ^π can achieve λ^*

System-dynamics under Π^*

$R_i(t)$: The *total number* of packets received by node i at t.

State Variables:

For each node $j \in V \setminus \{r\}$ define

$$X_j(t) \quad = \quad \min_{i:(i,j)\in E} [R_i(t) - R_j(t)] \text{ (\textit{Relative deficiency})}$$

Under any policy $\pi \in \Pi^*$ the state variables $X_j(t)$ satisfies a Lindley recursion.

Under Π^*, any algorithm stabilizing $\boldsymbol{X}(t)$ is a broadcast algorithm in a DAG.

Intuition : The state-vector $\boldsymbol{X}(t)$ mathematically corresponds to "queue-sizes" in the traditional queuing network.

Max-Weight Policy for Stabilizing $\boldsymbol{X}(t)$

To each edge $(i,j) \in E$, assign a non-negative weight $W_{ij}(t)$, where

$$W_{ij}(t) = \max\left(0, (X_j(t) - \sum_{k:j=i_t^*(k)} X_k(t))\right)$$

Choose a *max-weight* activation with weights $\boldsymbol{W}(t)$

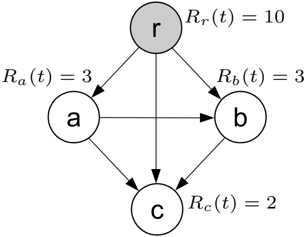
$$\boldsymbol{s}(t) \in \arg \max_{(s_e, e\in E)\in \mathcal{S}} \sum_{e\in E} c_e s_e W_e(t).$$

The policy π^* is an optimal broadcast policy for a DAG, i.e. for $\lambda < \lambda^*$

$$\liminf_{t\to\infty} \frac{R_i^{\pi^*}(t)}{t} = \lambda, \quad \forall i \in V \quad w.p.1$$

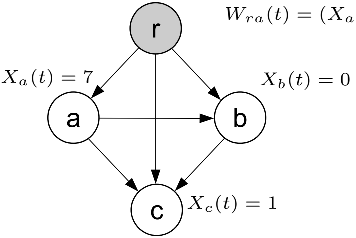
An Example

Step 1



$$
\begin{aligned}
&Q_{ra}(t) = 7 && K_r(t) = \{a\}\\
&Q_{rb}(t) = 7 && K_a(t) = \{b, c\}^*\\
&Q_{rc}(t) = 8 && K_b(t) = \{\emptyset\}\\
&Q_{ab}(t) = 0 && K_c(t) = \{\emptyset\}\\
&Q_{ac}(t) = 1 &&\\
&Q_{bc}(t) = 1 &&
\end{aligned}
$$

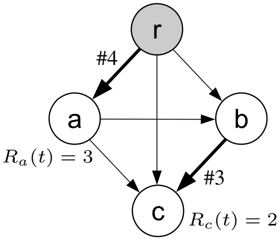
Step 2



$$
\begin{aligned}
W_{ra}(t) &= (X_a(t) - X_b(t) - X_c(t))^+ = 6\\
W_{rb}(t) &= (X_b(t))^+ = 0\\
W_{rc}(t) &= (X_c(t))^+ = 1\\
W_{ab}(t) &= (X_b(t))^+ = 0\\
W_{ac}(t) &= (X_c(t))^+ = 1\\
W_{bc}(t) &= (X_c(t))^+ = 1
\end{aligned}
$$

An Example

Step 3



$\mathbf{s}_1$: $W_{ra}(t) + W_{bc}(t) = 7$

$\mathbf{s}_2$: $W_{rb}(t) + W_{ac}(t) = 1$

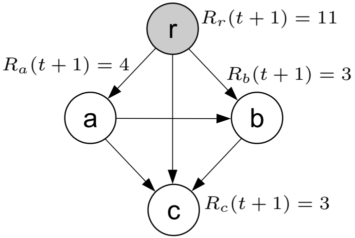
$\mathbf{s}_3$: $W_{rc}(t) + W_{ab}(t) = 1$

Choose the link-activation vector $\mathbf{s}_1$

Forward the next packet #4 on (r, a)

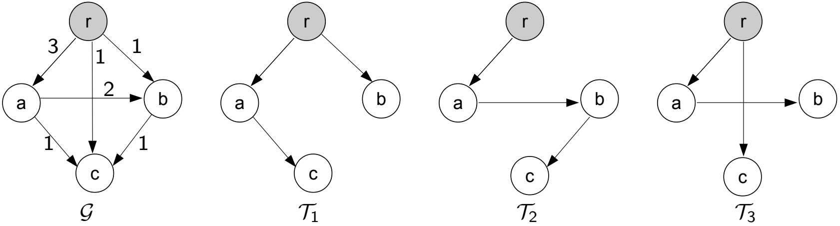
Forward the next packet #3 on (b, c)

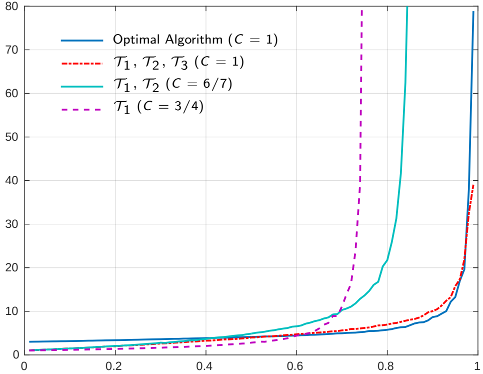
Step 4



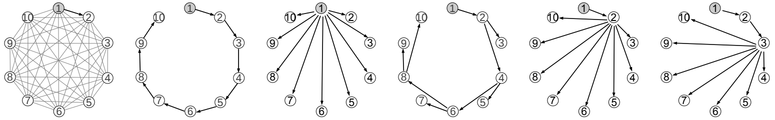
One packet arrives at the source

Simulation Topology and Spanning Trees





Delay Performance II



(a) Wireless topology (b) Tree $\mathcal{T}_1$ (c) Tree $\mathcal{T}_2$ (d) Tree $\mathcal{T}_3$ (e) Tree $\mathcal{T}_4$ (f) Tree $\mathcal{T}_5$

λ	Tree-based policy π_{tree} over the spanning trees: $\mathcal{T}_1$	$\mathcal{T}_1 \sim \mathcal{T}_2$	$\mathcal{T}_1 \sim \mathcal{T}_3$	$\mathcal{T}_1 \sim \mathcal{T}_4$	$\mathcal{T}_1 \sim \mathcal{T}_5$	Broadcast policy π^*
0.5	12.90	12.72	13.53	16.14	16.2	11.90
0.9	1.3×10^4	176.65	106.67	34.33	28.31	12.93
1.9	3.31×10^4	1.12×10^4	4.92×10^3	171.56	95.76	14.67
2.3	3.63×10^4	1.89×10^4	1.40×10^4	1.76×10^3	143.68	17.35
2.7	3.87×10^4	2.45×10^4	2.03×10^4	1.1×10^4	1551.3	20.08
3.1	4.03×10^4	2.86×10^4	2.51×10^4	1.78×10^4	9788.1	50.39

Table: Average delay performance of the tree-based policy π_{tree} over different subsets of spanning trees and the optimal broadcast policy π^*.

As can be seen, the algorithm has exceptionally good delay performance as compared to the tree-based broadcast.

Conclusions and Future Work

Broadcast is an efficient data transmission scheme.

All existing works require offline computation of spanning trees, which is impractical in large wireless networks

Derived an online throughput optimal broadcast algorithm for a wireless DAG.

Conclusions and Future Work

- Broadcast is an efficient data transmission scheme.
- All existing works require offline computation of spanning trees, which is impractical in large wireless networks
- **Derived** an online throughput optimal broadcast algorithm for a wireless DAG.

Open questions :

- To generalize this algorithm to arbitrary topology.
- To design an efficient algorithm which takes wireless broadcast advantage into account.

Questions ?