Optimal Distributed Broadcast Algorithms for Wireless DAGs

Abhishek Sinha

Joint work with Georgios Paschos, Chih-ping Li and Prof. Eytan Modiano,

Laboratory for Information and Decision Systems

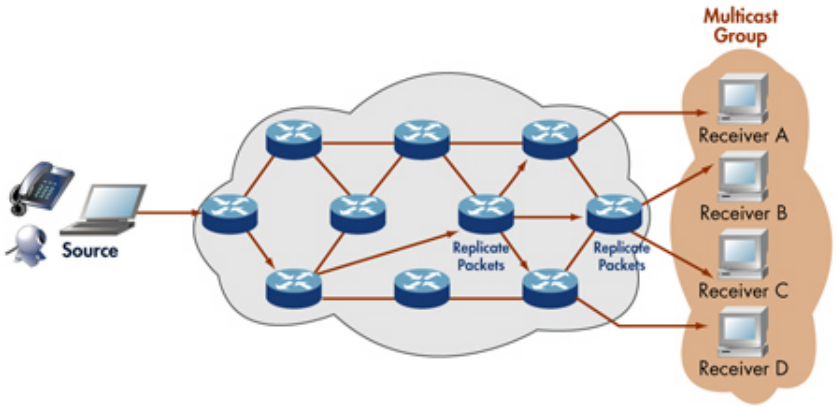
MIT

June 9, 2015

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.

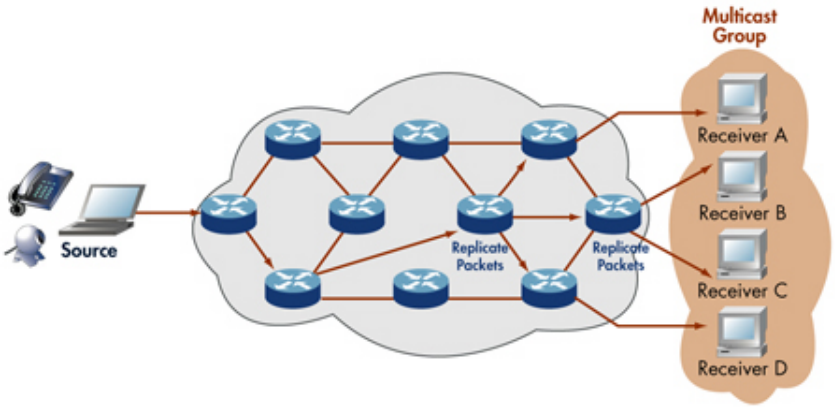


A wired network

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

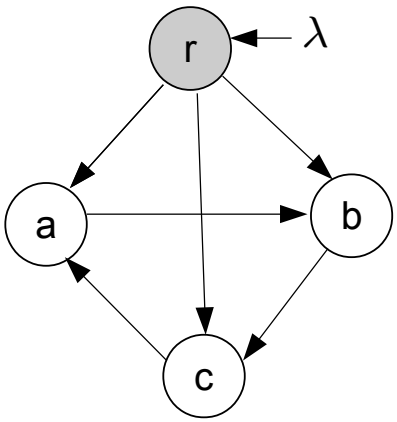
- In wireless settings, applications in military communications, disaster management over a mobile adhoc network (MANET).

System Model

- The network-topology is represented as a directed graph $\mathcal{G}(V, E)$.
- Time is slotted. Source $r \in V$ generates packets at rate $\lambda > 0$.
- Nodes can duplicate and relay packets to their neighbors.

System Model

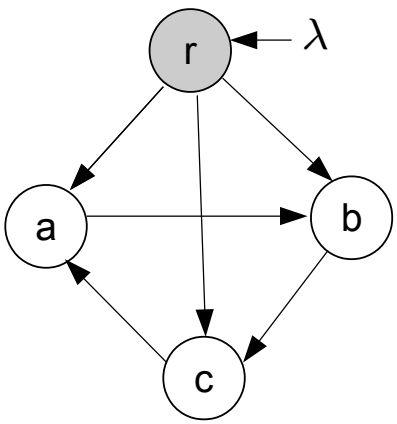
- The network-topology is represented as a directed graph $\mathcal{G}(V, E)$.
- Time is slotted. Source $r \in V$ generates packets at rate $\lambda > 0$.
- Nodes can duplicate and relay packets to their neighbors.
- Packet transmissions are subject to wireless interference constraints.



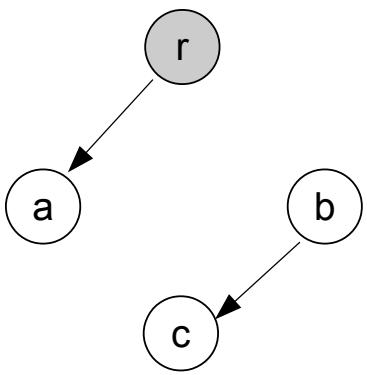
A wireless network

System Model

- The network-topology is represented as a directed graph $\mathcal{G}(V, E)$.
- Time is slotted. Source $r \in V$ generates packets at rate $\lambda > 0$.
- Nodes can duplicate and relay packets to their neighbors.
- Packet transmissions are subject to wireless interference constraints.



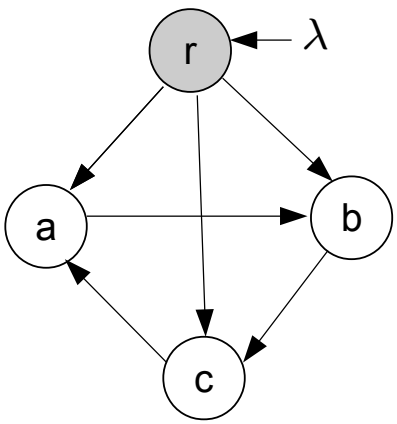
A wireless network



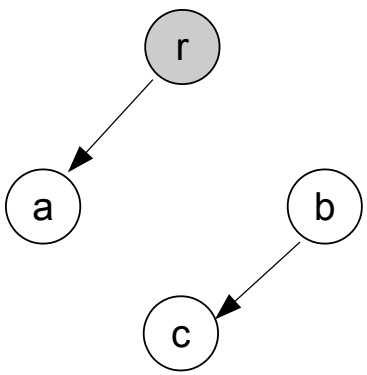
Activation $\boldsymbol{s}_1$

System Model

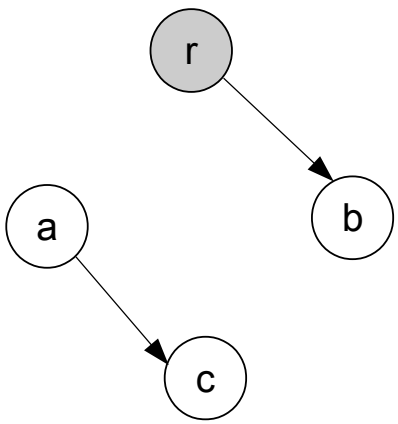
- The network-topology is represented as a directed graph $\mathcal{G}(V, E)$.
- Time is slotted. Source $r \in V$ generates packets at rate $\lambda > 0$.
- Nodes can duplicate and relay packets to their neighbors.
- Packet transmissions are subject to wireless interference constraints.



A wireless network



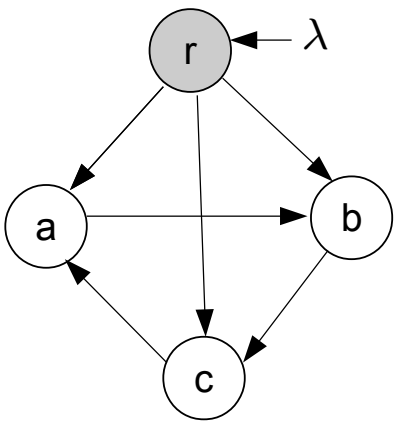
Activation $\boldsymbol{s}_1$



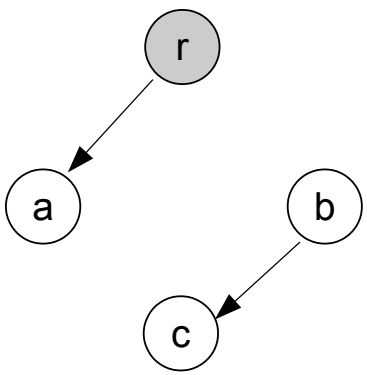
Activation $\boldsymbol{s}_2$

System Model

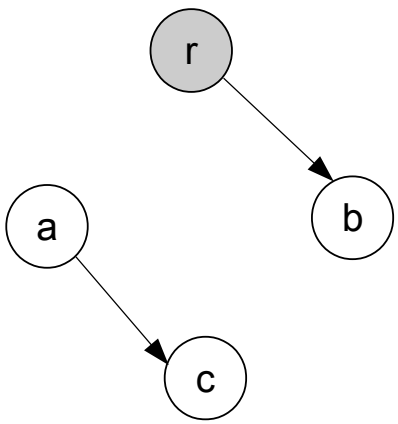
- The network-topology is represented as a directed graph $\mathcal{G}(V, E)$.
- Time is slotted. Source $r \in V$ generates packets at rate $\lambda > 0$.
- Nodes can duplicate and relay packets to their neighbors.
- Packet transmissions are subject to wireless interference constraints.



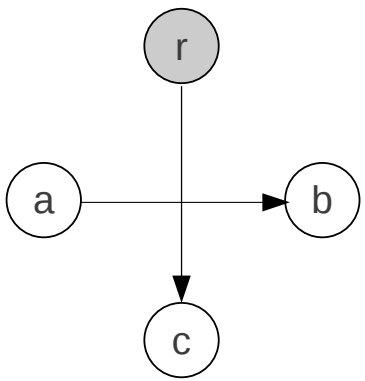
A wireless network



Activation s_1



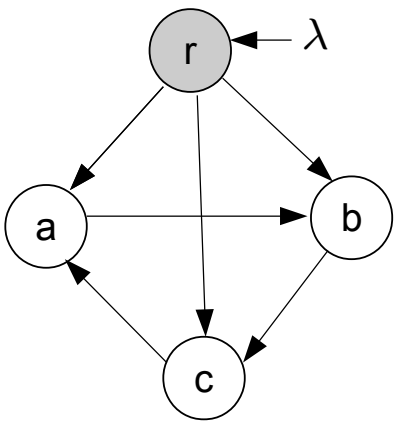
Activation s_2



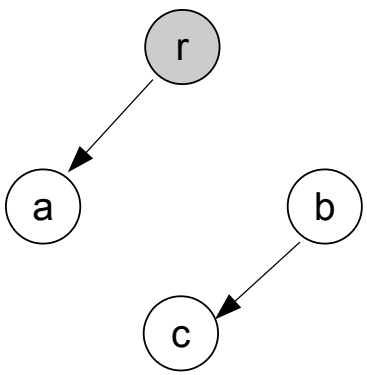
Activation s_3

System Model

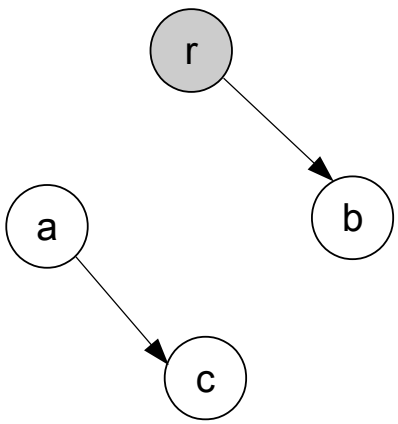
- The network-topology is represented as a directed graph $\mathcal{G}(V, E)$.
- Time is slotted. Source $r \in V$ generates packets at rate $\lambda > 0$.
- Nodes can duplicate and relay packets to their neighbors.
- Packet transmissions are subject to wireless interference constraints.



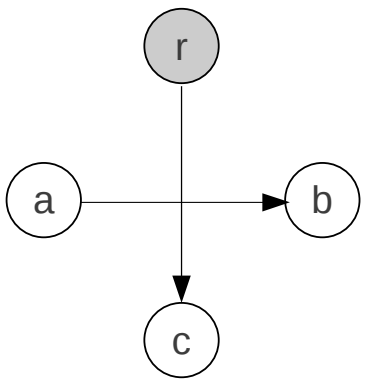
A wireless network



Activation s_1



Activation s_2



Activation s_3

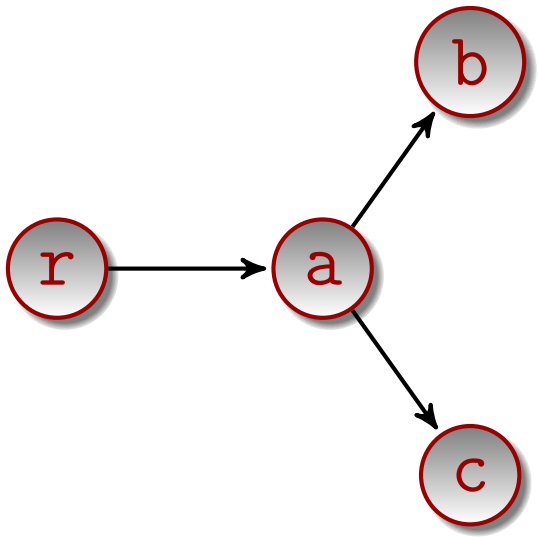
Challenge Traditional queuing theory does not apply due to packet duplications.

Naïve approach and its limitation

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

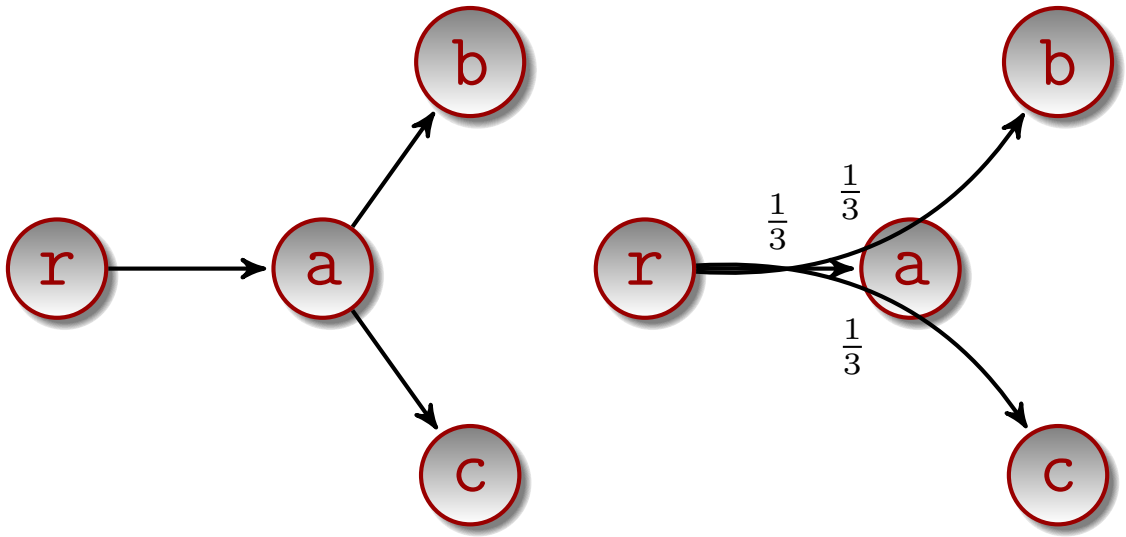
Naïve approach and its limitation

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



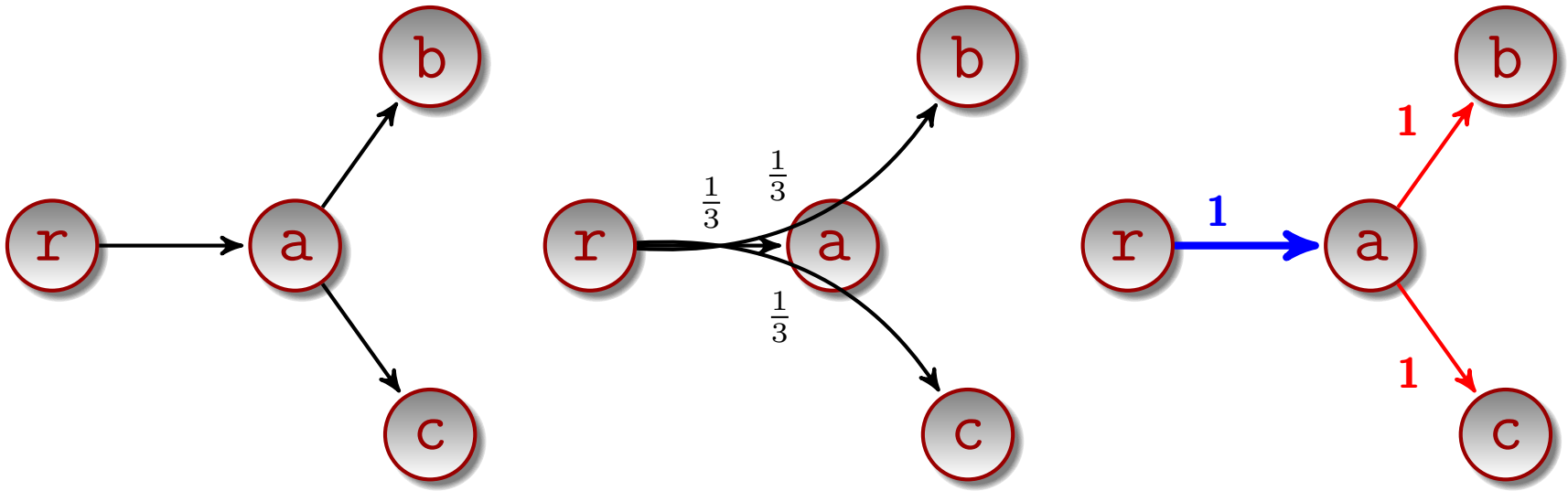
Naïve approach and its limitation

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



Naïve approach and its limitation

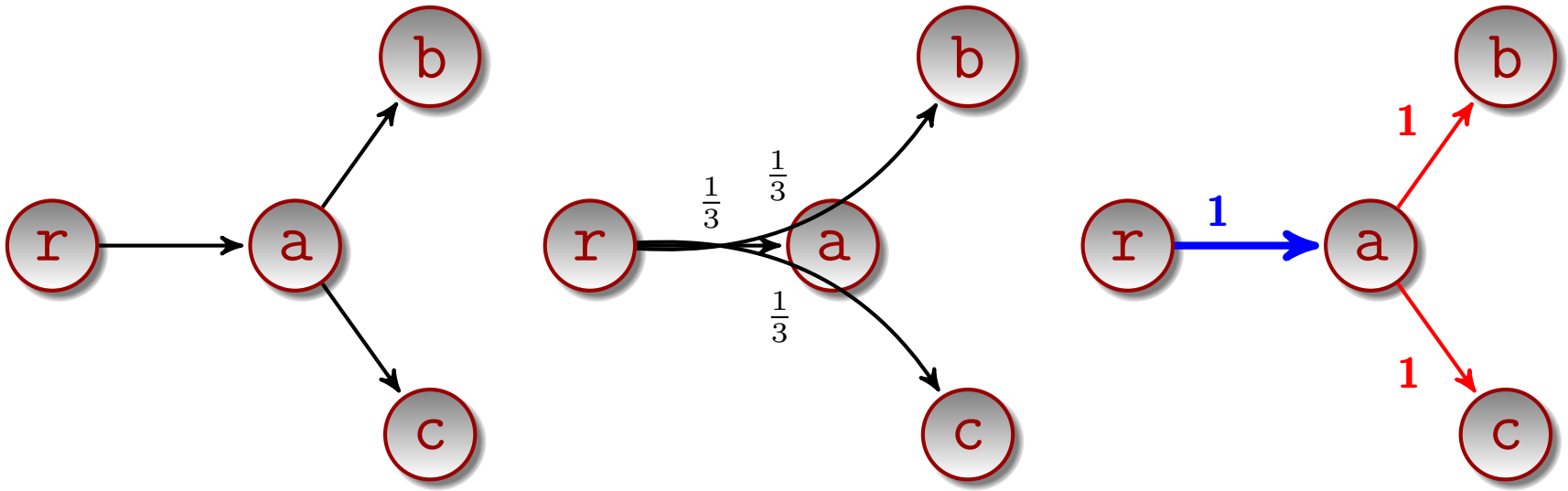
- Establish independent point-to-point unicast connections from source to all *other* nodes.



- Throughput limitation: redundant transmissions in the link r $\longrightarrow$ a. ✗

Naïve approach and its limitation

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



- Throughput limitation: redundant transmissions in the link r $\longrightarrow$ a. ✗
- Question How to route/split the packets in an intelligent way ?

The Broadcast Capacity λ^*

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

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

The Broadcast Capacity 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,

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

The Broadcast Capacity λ^*

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

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

Thus,

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

Edmond's Tree-Packing Theorem [1965]

- The above inequality is satisfied with equality.
- There exist λ^* edge-disjoint spanning trees to achieve the broadcast capacity.

The Broadcast Capacity λ^*

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

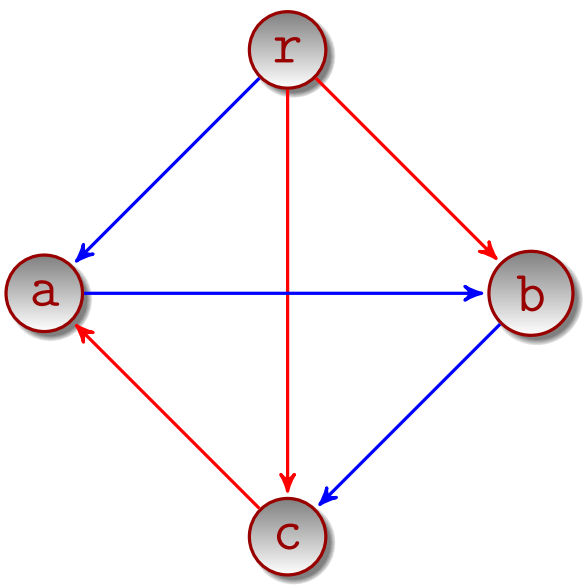
$$\lambda^* \leq \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Thus,

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

Edmond's Tree-Packing Theorem [1965]

- The above inequality is satisfied with equality.
- There exist λ^* edge-disjoint spanning trees to achieve the broadcast capacity.



The Broadcast Capacity λ^*

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

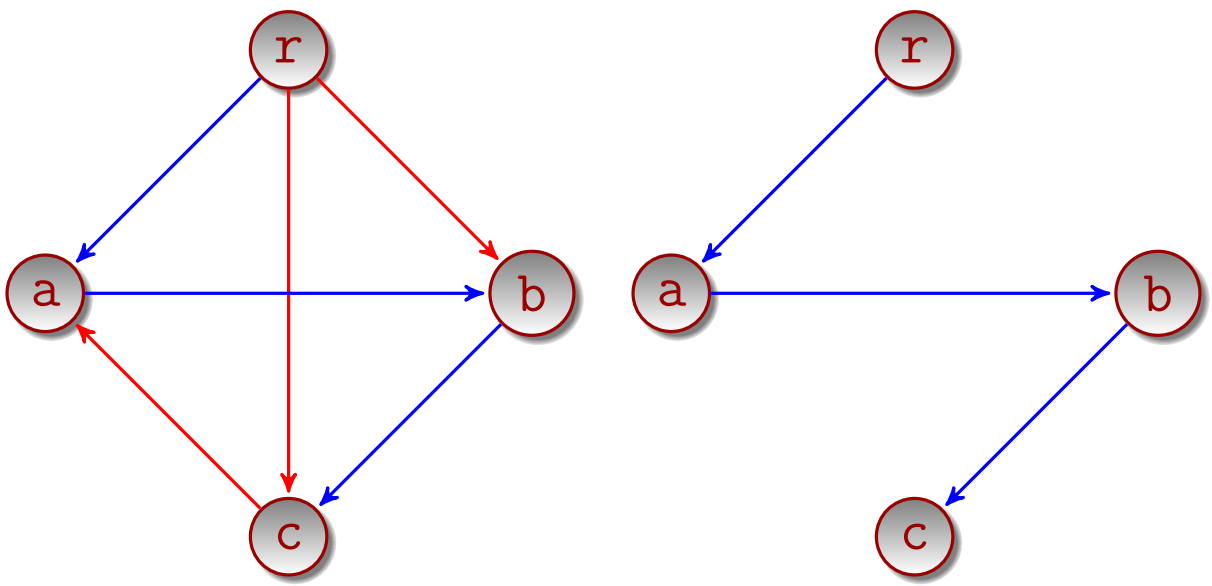
$$\lambda^* \leq \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Thus,

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

Edmond's Tree-Packing Theorem [1965]

- The above inequality is satisfied with equality.
- There exist λ^* edge-disjoint spanning trees to achieve the broadcast capacity.



The Broadcast Capacity λ^*

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

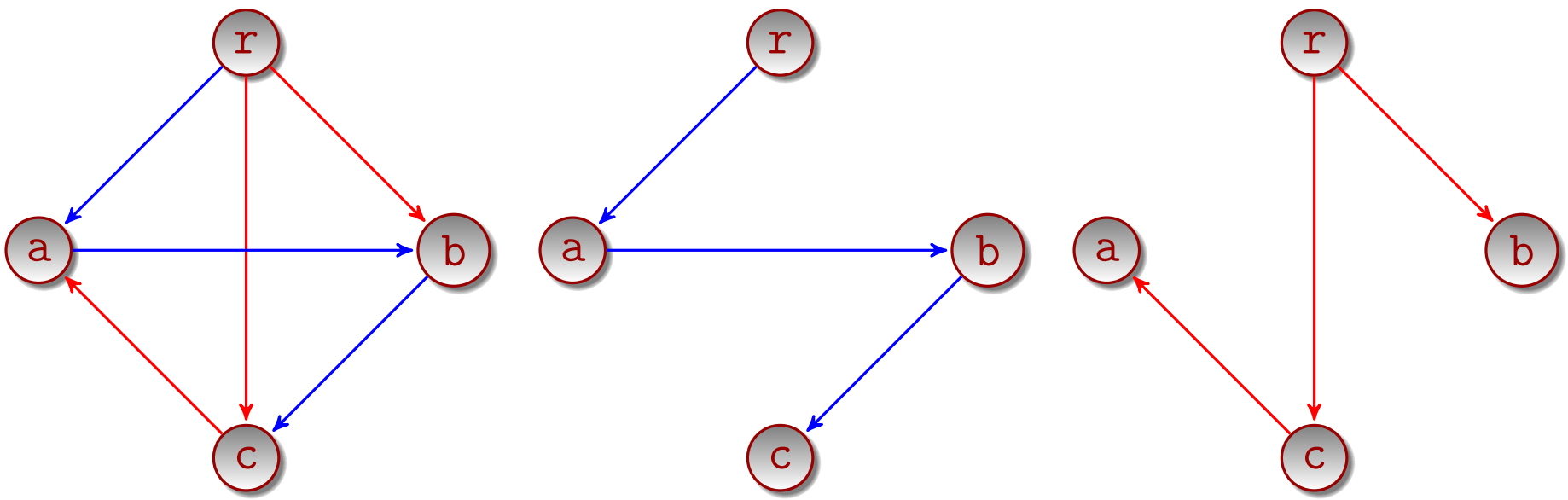
$$\lambda^* \leq \textsc{Max-Flow}(\mathtt{r} \to \mathtt{t})$$

Thus,

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

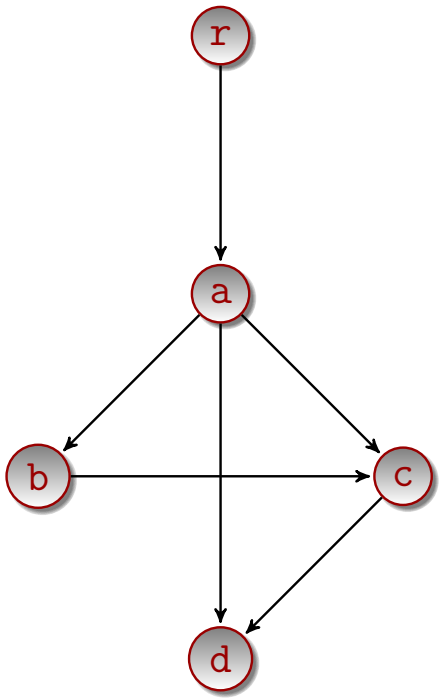
Edmond's Tree-Packing Theorem [1965]

- The above inequality is satisfied with equality.
- There exist λ^* edge-disjoint spanning trees to achieve the broadcast capacity.



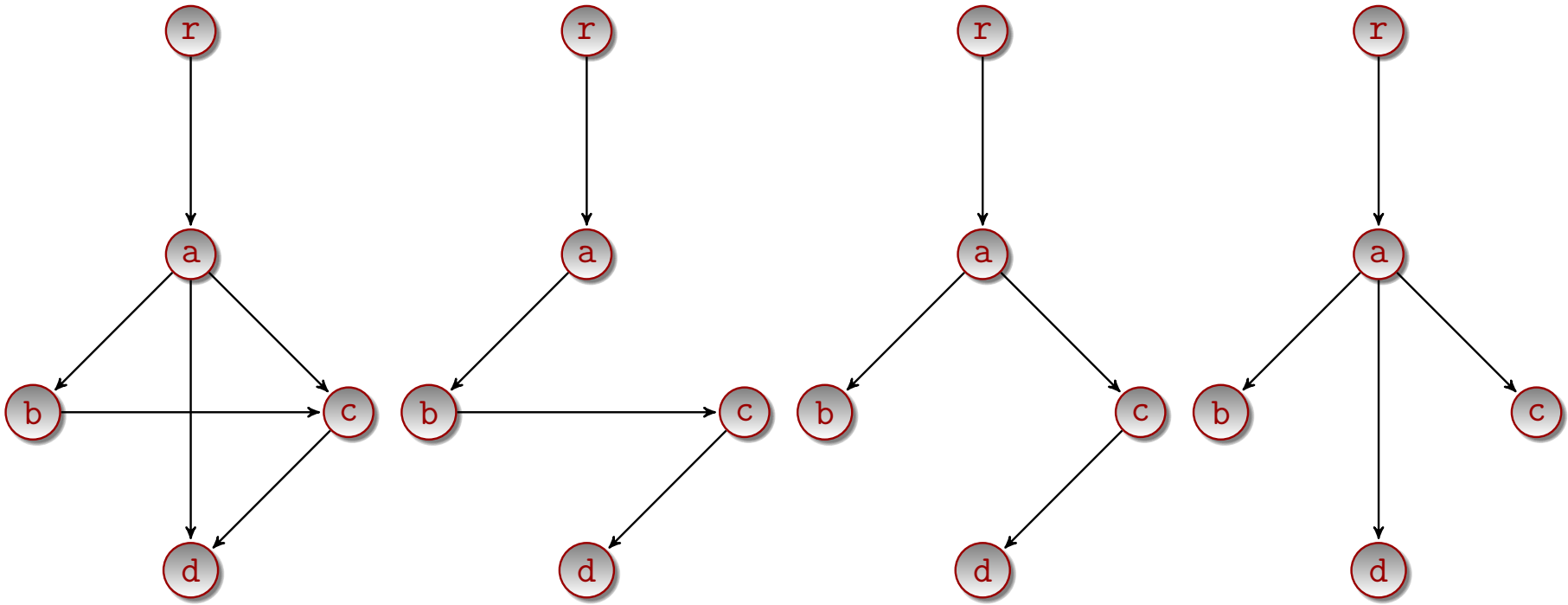
Previous Works

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



Previous Works

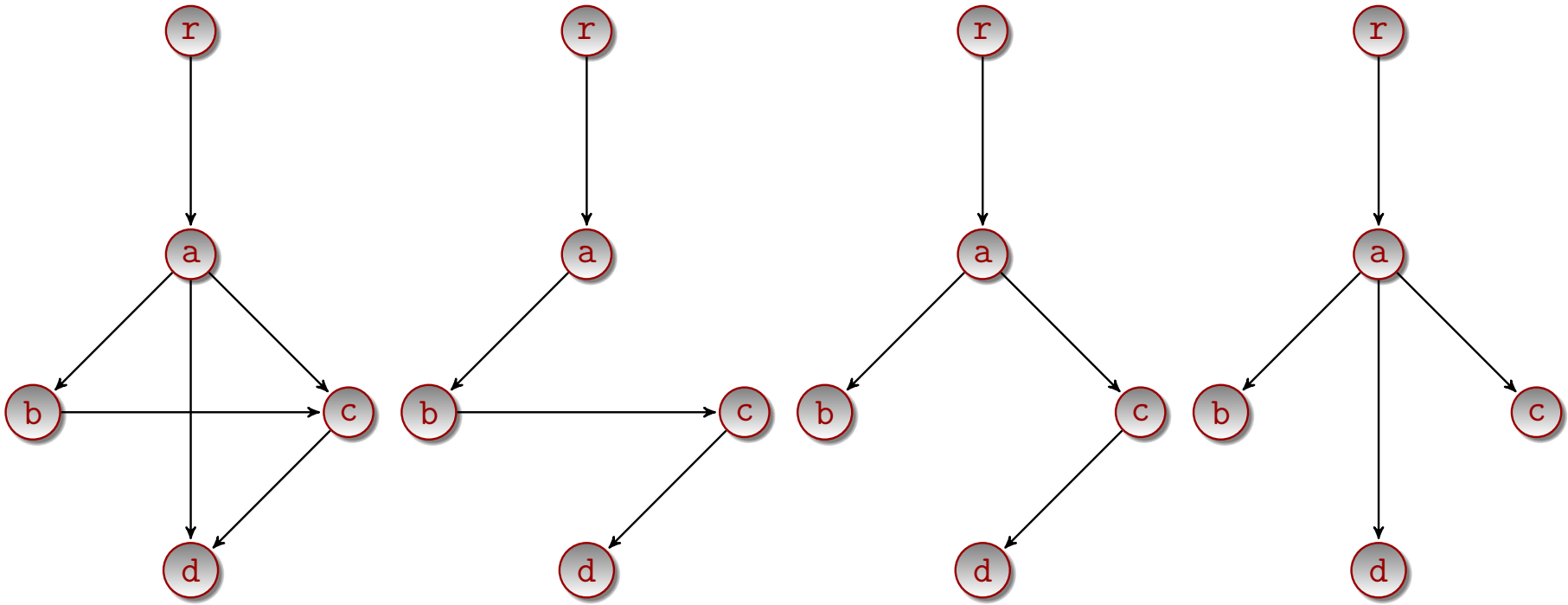
- Pre-computes the set of all spanning trees offline in wireline networks [Sarkar and Tassiulas, 2005].



  - Impractical for large networks and time-varying wireless networks. ✗
- Wireless case is studied with a fixed activation schedule [Towsley et al. 2008].

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].
- Our contribution is 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 $\pi(\mathcal{A})$: Activate a subset of links (e.g., a matching) subject to the underlying interference constraints.
- Packet Scheduling $\pi(\mathcal{S})$: Transmit packets over the set of activated links.

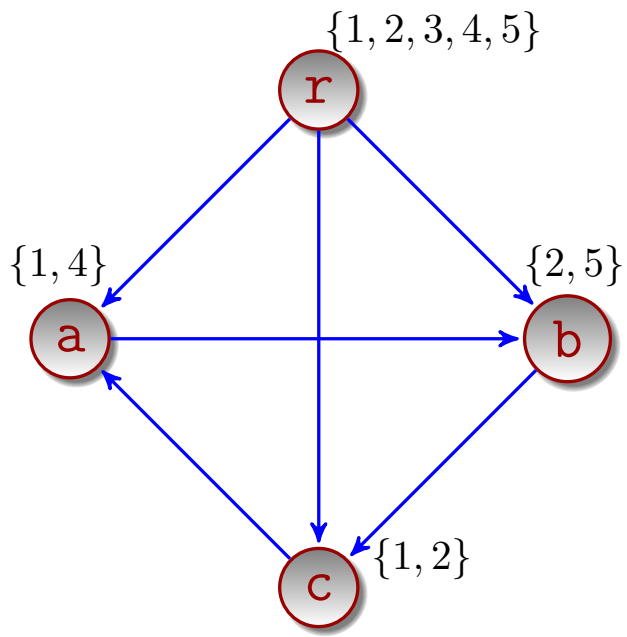
- An arbitrary $\pi(\mathcal{S})$ is hard to describe : exponential growth with packets!

Feasible Policy Space Π

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

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

- An arbitrary $\pi(\mathcal{S})$ is hard to describe : exponential growth with packets!

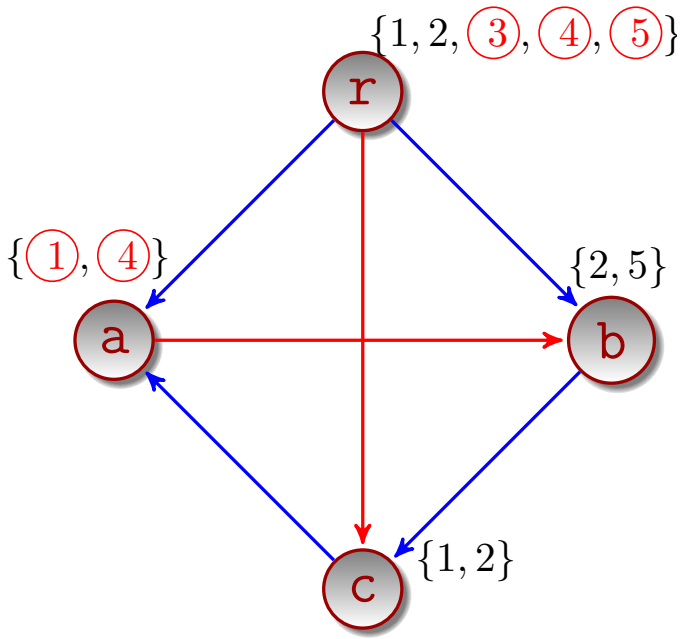


Feasible Policy Space Π

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

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

- An arbitrary $\pi(\mathcal{S})$ is hard to describe : exponential growth with packets!

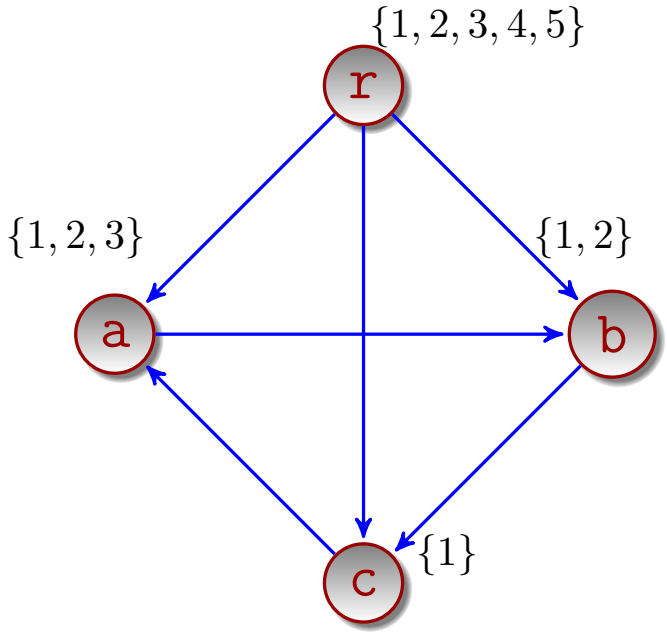


Policy-space $\Pi^* \subset \Pi^{\text{in-order}}$

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

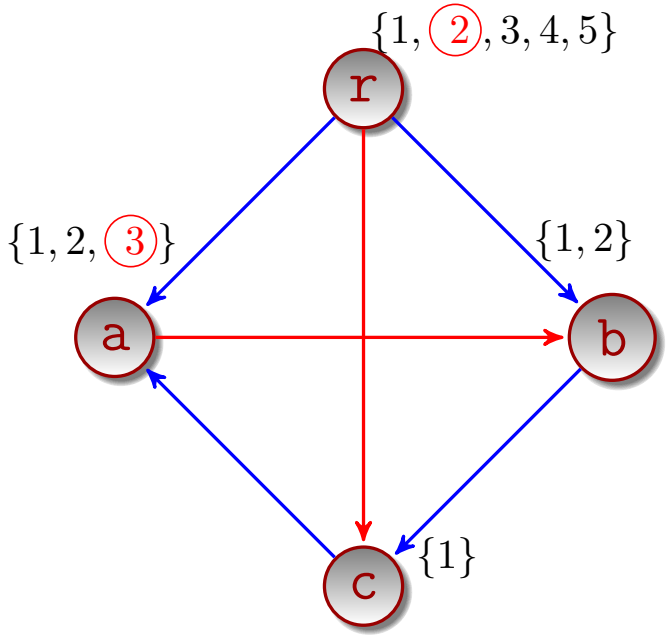
Policy-space $\Pi^* \subset \Pi^{\text{in-order}}$

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



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

- To simplify $\pi(\mathcal{S})$, 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

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

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

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

We show that the constrained policy space $\Pi^{\text{in-order}}$ contains a throughput optimal policy for DAGs.
To prove this fact, we consider a subspace $\Pi^* \subset \Pi^{\text{in-order}}$ defined as follows :

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

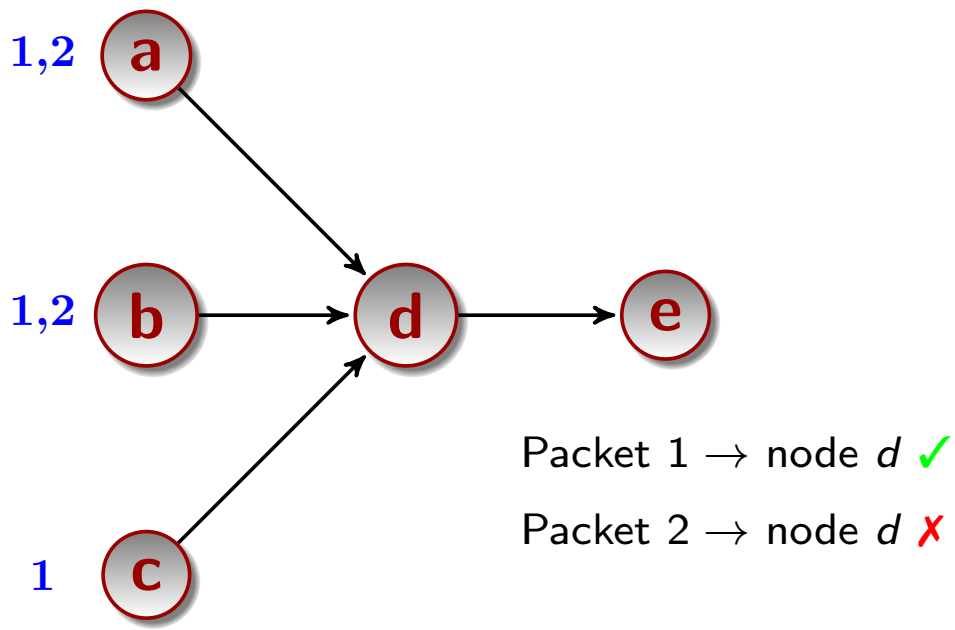
For all $\pi \in \Pi^*$, a packet p is eligible for transmission to node n iff all in-neighbors of node n contains the packet p.

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

We show that the constrained policy space $\Pi^{\text{in-order}}$ contains a throughput optimal policy for DAGs.
To prove this fact, we consider a subspace $\Pi^* \subset \Pi^{\text{in-order}}$ defined as follows :

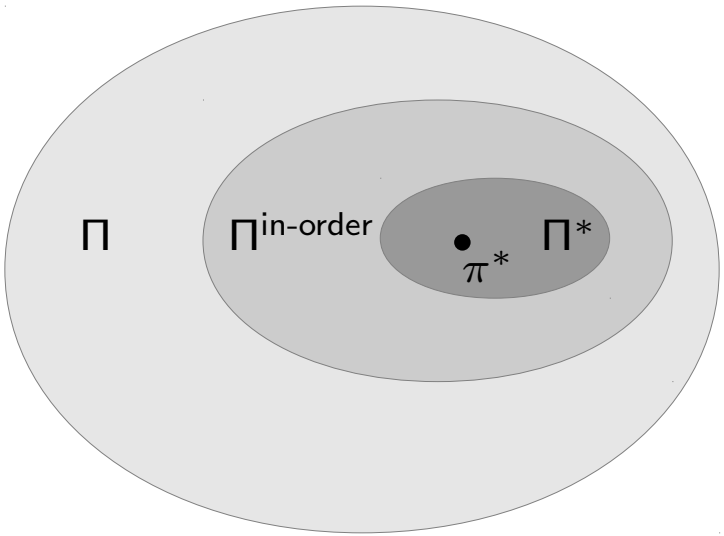
$\Pi^* \subset \Pi^{\text{in-order}} \subset \Pi$

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.



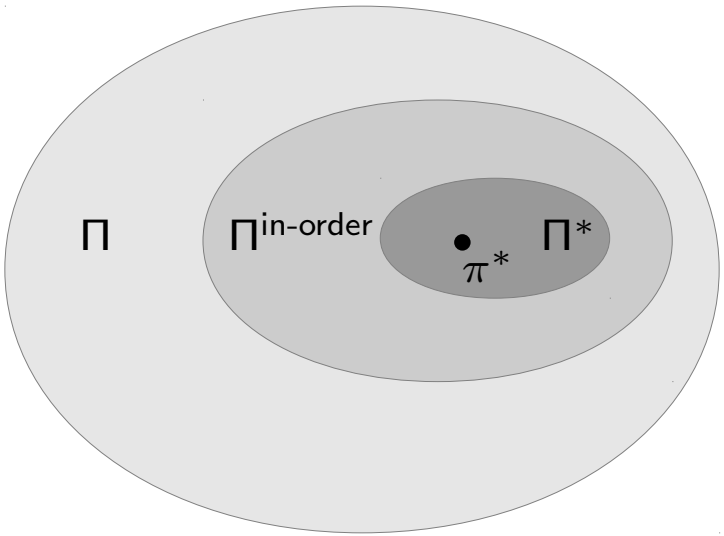
Policy hierarchies

All policies are un-restricted for link-activations ($\pi(\mathcal{A})$)



Policy hierarchies

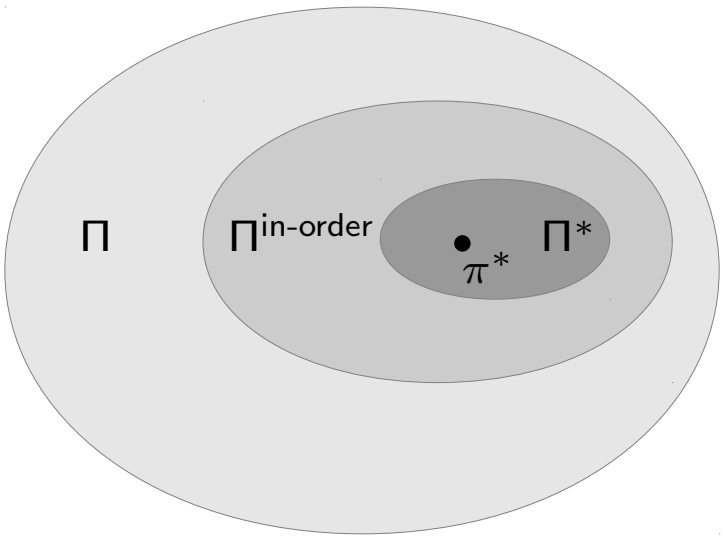
All policies are un-restricted for link-activations ($\pi(\mathcal{A})$)



Π: all policies that perform arbitrary packet-forwarding

Policy hierarchies

All policies are un-restricted for link-activations ($\pi(\mathcal{A})$)

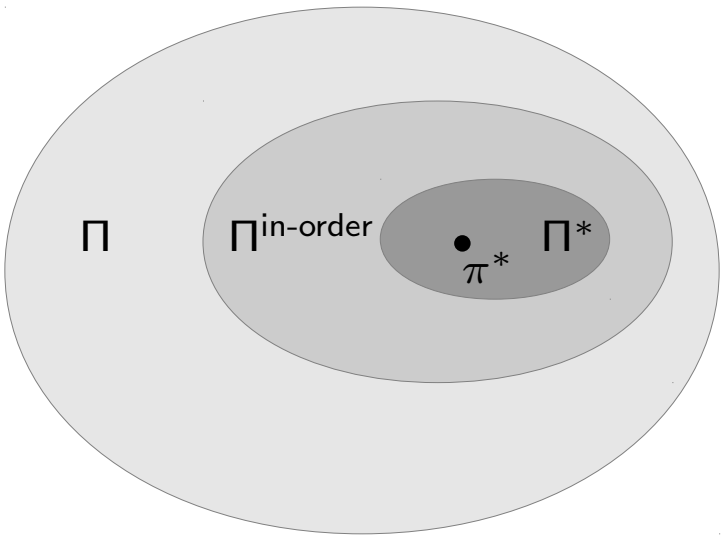


Π: all policies that perform arbitrary packet-forwarding

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

Policy hierarchies

All policies are un-restricted for link-activations ($\pi(\mathcal{A})$)



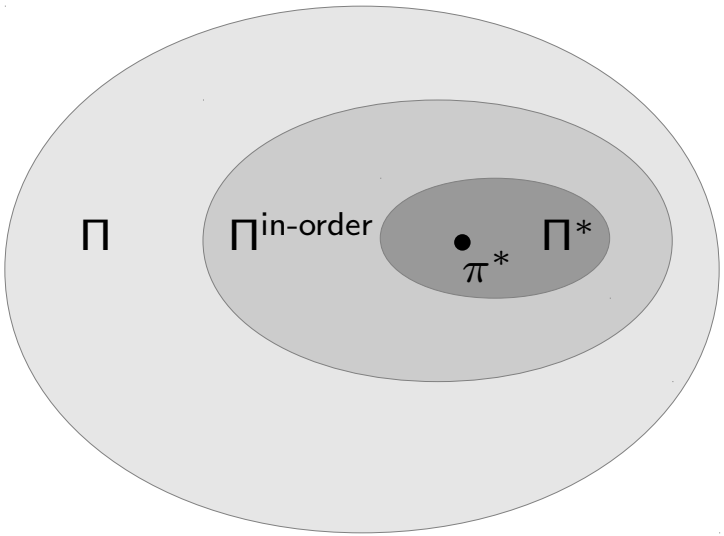
Π: all policies that perform arbitrary packet-forwarding

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

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

Policy hierarchies

All policies are un-restricted for link-activations ($\pi(\mathcal{A})$)



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!

System-dynamics under Π^*

State Variables:

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

$$X_j(t) \quad = \quad \min_{i:(i,j)\in E} \{R_i(t) - R_j(t), 0\} \quad \textit{(Relative deficiency)}$$

Theorem 1

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

System-dynamics under Π^*

State Variables:

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

$$X_j(t) \quad = \quad \min_{i:(i,j)\in E} \{R_i(t) - R_j(t), 0\} \quad \textit{(Relative deficiency)}$$

Theorem 1

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

Lemma 1

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 $\boldsymbol{X}(t)$

Consider a policy $\pi^* \in \Pi^*$, for which $\pi^*(\mathcal{A})$ is obtained by minimizing the Lyapunov function $L(\boldsymbol{X}(t)) = \sum_{j \in V \setminus \{r\}} X_j^2(t)$

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

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

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

$$\boldsymbol{\mu}^{\pi^*}(t) \in \arg\max_{\boldsymbol{\mu} \in \boldsymbol{C} \odot \mathcal{S}} \boldsymbol{\mu} \cdot \boldsymbol{W}(t) \tag{1}$$

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

Consider a policy $\pi^* \in \Pi^*$, for which $\pi^*(\mathcal{A})$ is obtained by minimizing the Lyapunov function $L(\boldsymbol{X}(t)) = \sum_{j \in V \setminus \{r\}} X_j^2(t)$

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

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

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

$$\boldsymbol{\mu}^{\pi^*}(t) \in \arg\max_{\boldsymbol{\mu} \in \boldsymbol{C} \odot \mathcal{S}} \boldsymbol{\mu} \cdot \boldsymbol{W}(t) \tag{1}$$

Theorem 3

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

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

Efficient Algorithm for Computing the Broadcast Capacity of a DAG

Theorem 4

The broadcast capacity of a wireless DAG is given by

$$\lambda^* = \lambda_{\text{DAG}} = \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e \, \beta_e$$

and there exists a *strongly polynomial time* algorithm for computing the capacity under primary interference constraint.

Efficient Algorithm for Computing the Broadcast Capacity of a DAG

Theorem 4

The broadcast capacity of a wireless DAG is given by

$$\lambda^* = \lambda_{\text{DAG}} = \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e \, \beta_e$$

and there exists a *strongly polynomial time* algorithm for computing the capacity under primary interference constraint.

Corollary

Activation of at most $|E| + 1$ are matchings are necessary to achieve the optimal capacity of any wireless DAG.

Extension of the DAG-broadcast algorithm to General topologies

Lemma 2

Π^* is strictly throughput sub-optimal for general network (even wired).

The basic DAG-broadcast algorithm can be extended to networks with general topology if we consider K classes where each class follow the policy Π^*.

Extension of the DAG-broadcast algorithm to General topologies

Lemma 2

Π^* is strictly throughput sub-optimal for general network (even wired).

The basic DAG-broadcast algorithm can be extended to networks with general topology if we consider K classes where each class follow the policy Π^*.

- Each class corresponds to a total ordering of the vertices.
- We choose them uniformly at random from the set of all permutations of vertices.

Extension of the DAG-broadcast algorithm to General topologies

Lemma 2

Π^* is strictly throughput sub-optimal for general network (even wired).

The basic DAG-broadcast algorithm can be extended to networks with general topology if we consider K classes where each class follow the policy Π^*.

- Each class corresponds to a total ordering of the vertices.
- We choose them uniformly at random from the set of all permutations of vertices.

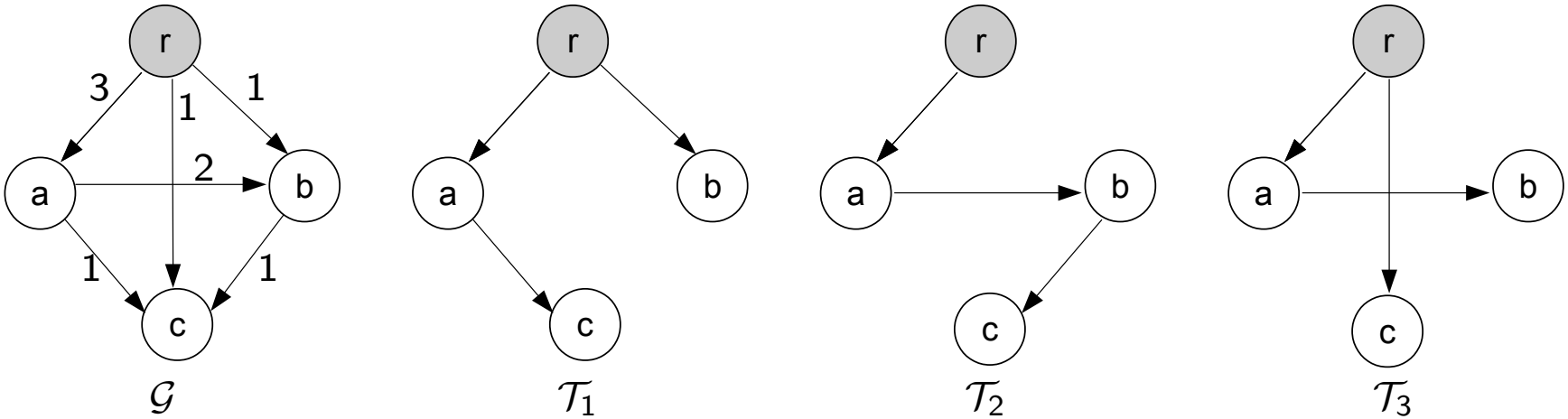
Theorem 5: Achievable rates with multiple classes

The multiclass broadcast algorithm with K classes supports a broadcast rate of

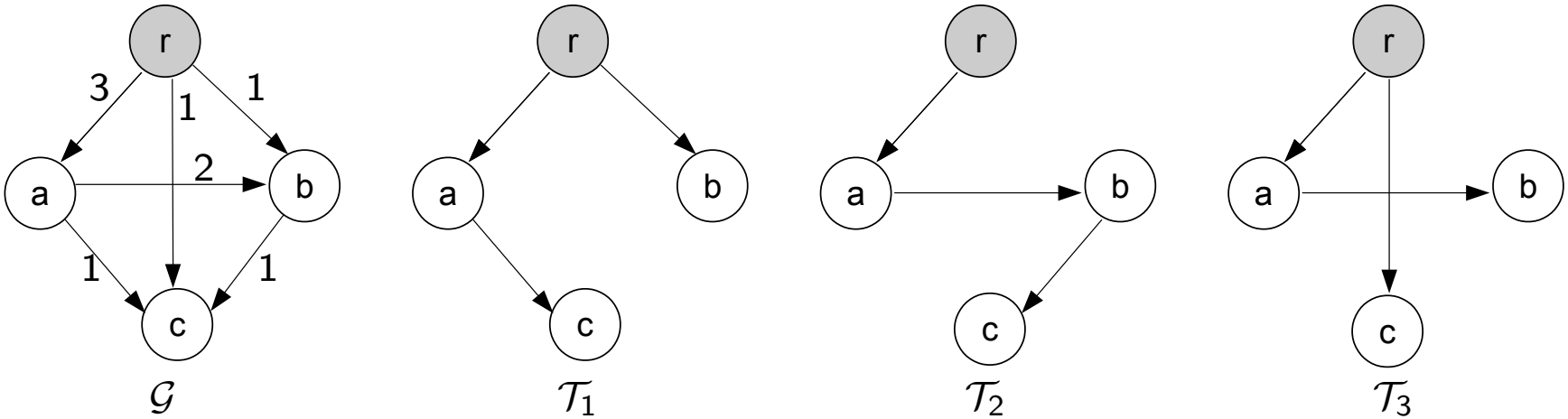
$$\lambda^K = \max_{\sum_k \boldsymbol{\beta}^k \in \mathrm{conv}(\mathcal{S})} \sum_{k=1}^{K} \min_{j \neq r} \sum_i c_{ij} \beta_{ij}^k \tag{2}$$

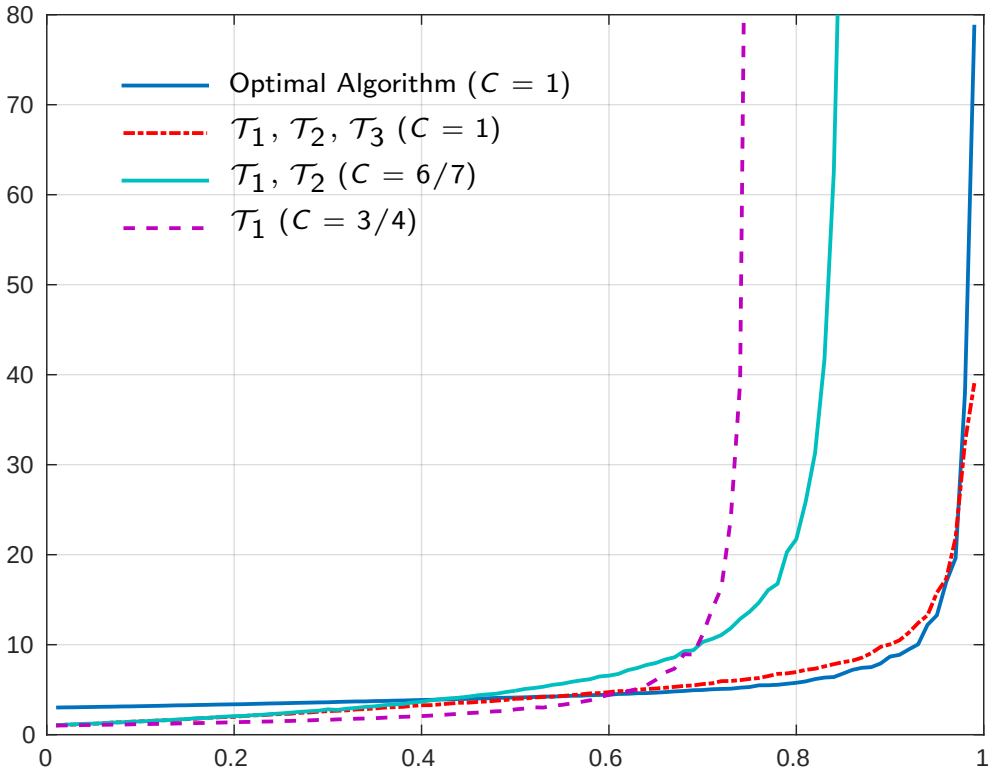
where we use the convention that $\beta_{ij}^k = 0$ if $(i,j) \notin E^{(k)}$.

Simulation Topology and Spanning Trees

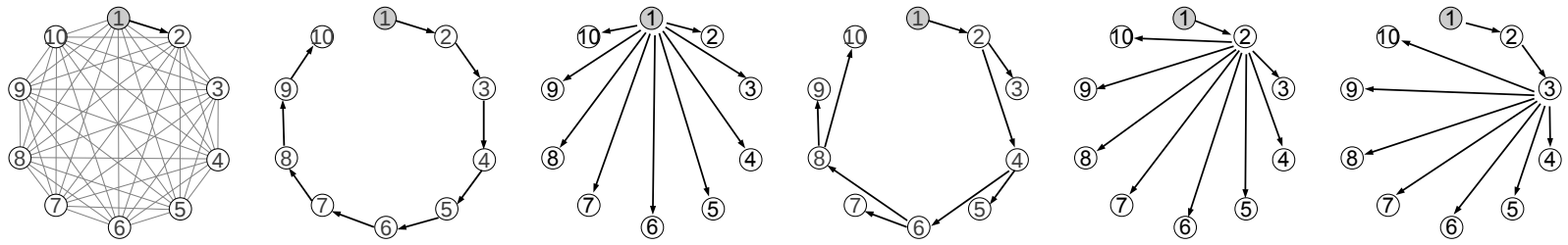


Simulation Topology and Spanning Trees





Delay Performance II



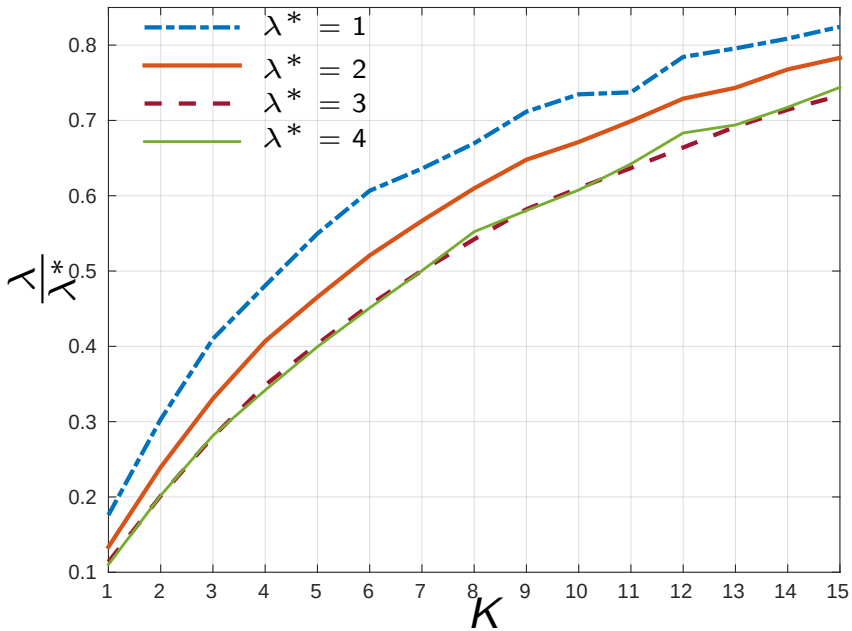
(a) Wireless topology (b) Tree $\mathcal{T}_1$ (c) Tree $\mathcal{T}_2$ (d) Tree $\mathcal{T}_3$ (e) Tree $\mathcal{T}_4$ (f) Tree $\mathcal{T}_5$

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, our algorithm has exceptionally good delay performance as compared to the tree-based broadcast.

Multiclass Simulation



Fraction of optimal broadcast rate $\frac{\lambda}{\lambda^*}$ achievable by the multiclass broadcast algorithm with randomly chosen K classes for randomly generated wired networks with $N = 10$ nodes.

Conclusions and Future Work

- Broadcast is an efficient data transmission scheme.

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

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
- We have 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
- We have 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 ?