

Throughput-Optimal Broadcast on Directed Acyclic Graphs

Abhishek Sinha*, Georgios Paschos[†], Chih-ping Li[‡], and Eytan Modiano*

*Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139

[†]Mathematical and Algorithmic Sciences Lab France Research Center, Huawei Technologies Co., Ltd.

[‡]Qualcomm Research, San Diego, CA

Email: *sinha@mit.edu, [†]georgios.paschos@huawei.com, [‡]cpli@qti.qualcomm.com, *modiano@mit.edu

Abstract—We study the problem of broadcasting packets in wireless networks. At each time slot, a network controller activates non-interfering links and forwards packets to all nodes at a common rate; the maximum rate is referred to as the broadcast capacity of the wireless network. Existing policies achieve the broadcast capacity by balancing traffic over a set of spanning trees, which are difficult to maintain in a large and time-varying wireless network. We propose a new dynamic algorithm that achieves the broadcast capacity when the underlying network topology is a directed acyclic graph (DAG). This algorithm utilizes local queue-length information, does not use any global topological structures such as spanning trees, and uses the idea of in-order packet delivery to all network nodes. Although the in-order packet delivery constraint leads to degraded throughput in cyclic graphs, we show that it is throughput optimal in DAGs and can be exploited to simplify the design and analysis of optimal algorithms. Our simulation results show that the proposed algorithm has superior delay performance as compared to tree-based approaches.

I. INTRODUCTION

Broadcast refers to the fundamental network functionality of delivering data from a source node to all other nodes. It uses packet replication and appropriate forwarding to eliminate unnecessary packet retransmissions. This is especially important in power-constrained wireless systems which suffer from interference and collisions. Broadcast applications include mission-critical military communications [1], live video streaming [2], and data dissemination in sensor networks [3].

The design of efficient wireless broadcast algorithms faces several challenges. Wireless channels suffer from interference, and a broadcast policy needs to activate non-interfering links at any time. Wireless network topologies undergo frequent changes, so that packet forwarding decisions must be made in an adaptive fashion. Existing dynamic multicast algorithms that balance traffic over spanning trees [4] may be used for broadcasting, since broadcast is a special case of multicast.

This work was supported by NSF Grant CNS-1217048, ONR Grant N00014-12-1-0064, and ARO MURI Grant W911NF-08-1-0238

[†] The work of G. Paschos was done while he was at MIT and affiliated with CERTH-ITI, and it was supported in part by the WiNC project of the Action: Supporting Postdoctoral Researchers, funded by national and Community funds (European Social Fund).

[‡] The work of C.p.Li was done when he was a Postdoctoral scholar at LIDS, MIT.

These algorithms, however, are not suitable for wireless networks because enumerating all spanning trees is computationally complex and needs to be performed repeatedly when the network topology changes.

In this paper, we study the fundamental performance of broadcasting packets in wireless networks. We consider a time-slotted system. At every slot, a scheduler decides which wireless links to activate and which packets to forward on activated links, so that all nodes receive packets at a common rate. The broadcast capacity is the maximum common reception rate of distinct packets over all scheduling policies. We then design optimal wireless broadcast algorithms without the use of spanning trees. To the best of our knowledge, there exists no capacity-achieving scheduling policy for wireless broadcast without spanning trees.

We begin by considering a rich class of scheduling policies Π that perform arbitrary link activations and packet forwarding, and characterize the broadcast capacity over this policy class Π . We impose two additional constraints that improve the understanding of the problem. First, we consider the subclass of policies $\Pi^{\text{in-order}} \subset \Pi$ that enforce the in-order delivery of packets. Second, we focus on the subset of policies $\Pi^* \subset \Pi^{\text{in-order}}$ that allows the reception of a packet by a node only if all its incoming neighbors have received the packet. It is intuitively clear that the policies in the more structured class Π^* are easier to describe and analyze, but may yield degraded throughput performance. We show the surprising result that when the underlying network topology is a directed acyclic graph (DAG), there is a control policy $\pi^* \in \Pi^*$ that achieves the broadcast capacity. In contrast, there exists a cyclic network in which no control policy in Π^* or $\Pi^{\text{in-order}}$ can achieve the broadcast capacity.

To enable the design of the optimal broadcast policy, we establish a *queue-like dynamics* for the system state, which is represented by relative packet deficit. This is non-trivial for the broadcast problem because explicit queueing structure is difficult to maintain in the network due to packet replication. As a result, achieving the broadcast capacity reduces to finding a scheduling policy that stabilizes the system states using the drift analysis [5], [6].

In this paper, our contributions include:

- We define the broadcast capacity of a wireless network

and show that it is characterized by an edge-capacitated graph $\widehat{\mathcal{G}}$ that arises from optimizing the time-averages of link activations. For integral-capacitated DAGs, the broadcast capacity is determined by the minimum in-degree of the graph $\widehat{\mathcal{G}}$, which is equal to the maximal number of edge-disjoint spanning trees.

- We design a dynamic algorithm that utilizes local queue-length information to achieve the broadcast capacity of a wireless DAG network. This algorithm does not rely on spanning trees, has small computational complexity, and is suitable for mobile networks with time-varying topology.
- We demonstrate the superior delay performance of our algorithm, as compared to centralized tree-based algorithm [4], via numerical simulations.

In the literature, a simple method for wireless broadcast is to use packet flooding [7]. The flooding approach, however, leads to redundant transmissions and collisions, known as *broadcast storm* [8]. In the wired domain, it has been shown that forwarding *useful* packets at random is optimal for broadcast [9]; this approach does not extend to the wireless setting due to interference and the need for scheduling [10]. Broadcast on wired networks can also be done using network coding [11], [12]. However, efficient link activation under network coding remains an open problem.

The rest of the paper is organized as follows. Section II introduces the wireless network model. In Section III, we define the broadcast capacity of a wireless network and provide a useful upper bound from a fundamental cut-set bound. In Section IV, we propose a dynamic broadcast policy that achieves the broadcast capacity in a DAG. Simulation results are presented in Section V. Due to space limitations, we provide a subset of the proofs in the Appendix; see the technical report [13] for the complete proofs.

II. THE WIRELESS NETWORK MODEL

We consider a wireless network that is represented by a directed graph $\mathcal{G} = (V, E, \mathbf{c}, \mathcal{S})$, where V is the set of nodes, E is the set of directed point-to-point links, $\mathbf{c} = (c_{ij})$ denotes the capacity of links $(i, j) \in E$, \mathcal{S} is the set of all feasible link-activation vectors, and $\mathbf{s} = (s_e, e \in E) \in \mathcal{S}$ is a binary vector indicating that the links e with $s_e = 1$ can be activated simultaneously. The structure of the activation set \mathcal{S} depends on the interference model. Under the primary interference constraint, the set \mathcal{S} consists of all binary vectors corresponding to matchings of the underlying graph \mathcal{G} [14], see Fig. 1. In the case of a wired network, \mathcal{S} is the set of all binary vectors since there is no interference. In this paper, we allow an arbitrary link-activation set \mathcal{S} , which captures different wireless interference models. Let $r \in V$ be the source node at which the broadcast traffic is generated. We consider a time-slotted system. In slot t , the number of packets generated at the node r is denoted by $A(t)$, where $A(t)$ is i.i.d. over slots with mean λ . These packets need to be delivered to all nodes in the wireless network.

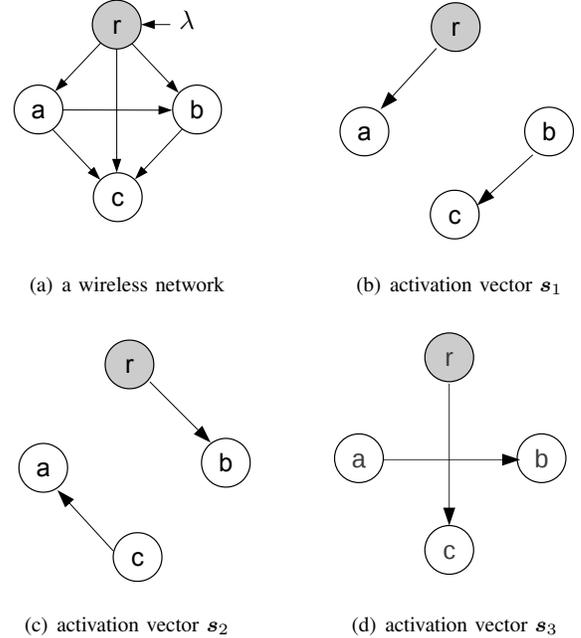


Fig. 1: A wireless network and its three feasible link activations under the primary interference constraint.

III. WIRELESS BROADCAST CAPACITY

Intuitively, the network supports a broadcast rate λ if there exists a scheduling policy under which all network nodes can receive distinct packets at rate λ . The broadcast capacity is the maximally supportable broadcast rate in the network. Formally, we consider a class Π of scheduling policies where each policy $\pi \in \Pi$ is a sequence of actions $\{\pi_t\}_{t \geq 1}$ taken in every slot t . Each action π_t consists of two operations: (i) the scheduler activates a subset of links by choosing a feasible activation vector $\mathbf{s} \in \mathcal{S}$; (ii) each node i forwards a subset of packets (possibly empty) to node j over an activated link (i, j) , subject to the link capacity constraint. The class Π includes policies that use all past and future network information, and may forward any subset of packets over a link.

To introduce the notion of broadcast capacity, we define the random variable $R_i^\pi(t)$ to be the number of distinct packets received by node $i \in V$ from the beginning of time up to time t , under a policy $\pi \in \Pi$. The time average $\lim_{T \rightarrow \infty} R_i^\pi(T)/T$ is the rate of distinct packets received at node i .

Definition 1. A policy π is called a “broadcast policy of rate λ ” if all nodes receive distinct packets at rate λ , i.e.,

$$\lim_{T \rightarrow \infty} \frac{1}{T} R_i^\pi(T) = \lambda, \quad \text{for all } i \in V, \text{ w. p. 1,} \quad (1)$$

where λ is the packet arrival rate at the source node r .¹

¹We can use the following more rigorous condition in (1):

$$\min_{i \in V} \liminf_{T \rightarrow \infty} \frac{1}{T} R_i^\pi(T) = \lambda, \quad \text{w. p. 1,} \quad (2)$$

under which all results in this paper still hold.

Definition 2. The broadcast capacity λ^* of a wireless network is the supremum of all arrival rates λ for which there exists a broadcast policy π of rate λ , $\pi \in \Pi$.

A. An upper bound on broadcast capacity λ^*

We characterize the broadcast capacity λ^* of a wireless network by providing a useful upper bound. This upper bound is understood as a necessary cut-set bound of an associated edge-capacitated graph that reflects the time-average behavior of the wireless network. We provide an intuitive explanation of the bound that will be formalized in Theorem 1 as follows.

Fix a policy $\pi \in \Pi$. Let β_e^π be the fraction of time link $e \in E$ is activated under π ; that is, we define the vector

$$\beta^\pi = (\beta_e^\pi, e \in E) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \mathbf{s}^\pi(t), \quad (3)$$

where $\mathbf{s}^\pi(t)$ is the link-activation vector under policy π in slot t (assuming all limits exist). The average flow rate over a link e under policy π is upper bounded by the product of the link capacity and the fraction of time the link e is activated, i.e., $c_e \beta_e^\pi$. It is convenient to define an edge-capacitated graph $\hat{\mathcal{G}} = (V, E, (\hat{c}_e))$ associated with policy π , where each directed link $e \in E$ has capacity $\hat{c}_e = c_e \beta_e^\pi$; see Fig. 2 for an example of such an edge-capacitated graph. Next, we provide a bound on the broadcast capacity by maximizing the broadcast rate at each node on the graph $\hat{\mathcal{G}}$ over all feasible vectors β^π .

We define a *proper cut* U of the network graph \mathcal{G} (or $\hat{\mathcal{G}}$) as a proper subset of the node set V that contains the source node r . Define the link subset

$$E_U = \{(i, j) \in E \mid i \in U, j \notin U\}. \quad (4)$$

Since $U \subset V$, there exists a node $n \in V \setminus U$. Consider the throughput of node n under policy π . The max-flow min-cut theorem shows that the throughput of node n cannot exceed the total link capacity $\sum_{e \in E_U} c_e \beta_e^\pi$ across the cut U . Since the achievable broadcast rate λ^π of policy π is a lower bound on the throughput of all nodes, we have $\lambda^\pi \leq \sum_{e \in E_U} c_e \beta_e^\pi$. This inequality holds for all proper cuts U and we have

$$\lambda^\pi \leq \min_{U: \text{a proper cut}} \sum_{e \in E_U} c_e \beta_e^\pi. \quad (5)$$

Equation (5) holds for any policy $\pi \in \Pi$. Thus, the broadcast capacity λ^* of the wireless network satisfies

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

where the last inequality holds because the vector β^π associated with any policy $\pi \in \Pi$ lies in the convex hull $\text{conv}(\mathcal{S})$ of the activation set \mathcal{S} . The next theorem formalizes the above characterization of the broadcast capacity λ^* of a wireless network.

Theorem 1. The broadcast capacity λ^* of a wireless network

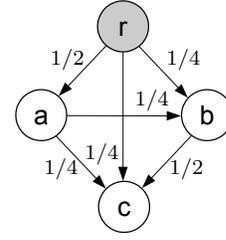


Fig. 2: The edge-capacitated graph $\hat{\mathcal{G}}$ for the wireless network with unit link capacities in Fig. 1 and under the time-average vector $\beta^\pi = (1/2, 1/4, 1/4)$. The link weights are the capacities $c_e \beta_e^\pi$. The minimum proper cut in this graph has value $1/2$ (when $U = \{r, a, c\}$ or $\{r, b, c\}$). An upper bound on the broadcast capacity is obtained by maximizing this value over all vectors $\beta^\pi \in \text{conv}(\mathcal{S})$.

under general interference constraints satisfies

$$\lambda^* \leq \max_{\beta \in \text{conv}(\mathcal{S})} \min_{U: \text{a proper cut}} \sum_{e \in E_U} c_e \beta_e. \quad (6)$$

Proof of Theorem 1: See the technical report [13]. ■

B. In-order packet delivery

Studying the performance of a general broadcast policy is difficult because packets are replicated across the network and may be received out of order. To avoid retransmissions of a packet, the nodes must keep track of the identity of the received packets, which complicates the system state—instead of the number of packets received, the system state is described by the subset of packets received at all nodes.

To simplify the system state, we focus on the subset of policies that enforce the *in-order packet delivery* constraint:

Constraint 1 (In-order packet delivery). A network node is allowed to receive a packet p only if all packets $\{1, 2, \dots, p-1\}$ have been received by that node.

We denote this policy subclass by $\Pi_{\text{in-order}}$. In-order packet delivery is useful in live media streaming applications [2], in which buffering out-of-order packets incurs increased delay that degrades video quality. In-order packet delivery greatly simplifies the network state space. Let $R_i(t)$ be the number of distinct packets received by node i by time t . For policies in $\Pi_{\text{in-order}}$, the set of received packets by time t at node i is $\{1, \dots, R_i(t)\}$. Therefore, the network state in slot t is completely represented by the vector $\mathbf{R}(t) = (R_i(t), i \in V)$.

Imposing the in-order packet delivery constraint potentially reduces the wireless broadcast capacity. It is of interest to study under what conditions in-order packet delivery can attain the broadcast capacity λ^* of the general policy class Π . The next lemma shows that there exists a cyclic network topology in which any broadcast policy enforcing the in-order packet delivery constraint is not throughput optimal.

Lemma 1. Let $\lambda_{\text{in-order}}^*$ be the broadcast capacity of the policy subclass $\Pi_{\text{in-order}} \subset \Pi$ that enforces in-order packet delivery.

There exists a network topology with directed cycles such that $\lambda^*_{in-order} < \lambda^*$.

Proof: See Appendix A.

C. Achieving broadcast capacity of a DAG

In the rest of the paper, we study optimal broadcast algorithms when the network topology is a DAG. For this, we simplify the upper bound (6) on the broadcast capacity λ^* in Theorem 1 in the case of a DAG. For each receiver node $v \neq r$, we consider the proper cut U_v that separates the network from node v :

$$U_v = V \setminus \{v\}. \quad (7)$$

Using these cuts $\{U_v, v \neq r\}$, we define another upper bound λ_{DAG} on the broadcast capacity λ^* as:

$$\begin{aligned} \lambda_{DAG} &\triangleq \max_{\beta \in \text{conv}(S)} \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e \beta_e \quad (8) \\ &\geq \max_{\beta \in \text{conv}(S)} \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e \geq \lambda^*, \end{aligned}$$

where the first inequality uses the subset relation $\{U_v, v \neq r\} \subseteq \{U: \text{ a proper cut}\}$ and the second inequality follows from Theorem 1. In Section IV, we will propose a dynamic policy that belongs to the policy class $\Pi_{in-order}$ and achieves any arrival rate that is less than λ_{DAG} . Combining this result with (8), we establish that the broadcast capacity of a DAG is

$$\begin{aligned} \lambda^* = \lambda_{DAG} &= \max_{\beta \in \text{conv}(S)} \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e \beta_e \\ &= \max_{\beta \in \text{conv}(S)} \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e, \quad (9) \end{aligned}$$

which is achieved by a broadcast policy that uses in-order packet delivery. In other words, imposing the in-order packet delivery constraint does not reduce the broadcast capacity when the network topology is a DAG.

IV. DAG BROADCAST ALGORITHM

We design an optimal broadcast policy for a wireless DAG. We start with imposing an additional constraint that leads to a new subclass of policies in which it is possible to describe the network dynamics by means of relative packet deficits, i.e., $R_i(t) - R_j(t)$, between nodes i and j . We analyze the dynamics of the minimum relative packet deficit at each node j , where the minimization is over all incoming neighbors of j . The minimum relative deficits play the role of virtual queues in the system, and we design a control policy that stabilizes them. The main result of this section is to show that this control policy achieves the broadcast capacity whenever the network topology is a DAG.

A. System state by means of packet deficits

We show in Section III-B that, in the policy class $\Pi_{in-order}$, the system state is represented by the vector $\mathbf{R}(t)$. To simplify the system dynamics further, we impose another constraint on $\Pi_{in-order}$ as follows. We say that node i is an *in-neighbor*

of node j if there exists a directed link $(i, j) \in E$ in the underlying graph \mathcal{G} .

■ **Constraint 2.** A packet p is eligible for transmission to node j in a slot only if all the in-neighbors of j have received packet p in previous slots.

We denote this new policy class by $\Pi^* \subseteq \Pi_{in-order}$.² Fig. 3 shows the relationship between different policy classes discussed in this paper.

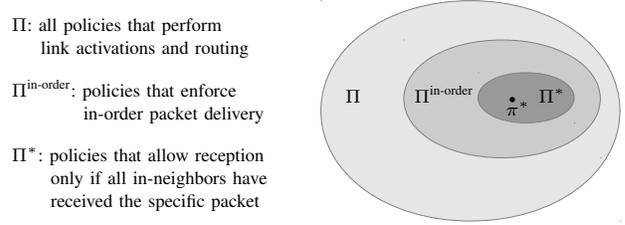


Fig. 3: The relationship between different policy classes.

The following properties of the system states $\mathbf{R}(t)$ under a policy in Π^* are useful.

Lemma 2. Let $In(j)$ denote the set of in-neighbors of a node j in the network. Under any policy $\pi \in \Pi^*$, we have:

- (1) $R_j(t) \leq \min_{i \in In(j)} R_i(t)$ for all nodes $j \neq r$.
- (2) The collection of packets that are eligible to be received by a node $j \neq r$ from its in-neighbors in slot t is

$$\{p \mid R_j(t) + 1 \leq p \leq \min_{i \in In(j)} R_i(t)\}.$$

We define the *packet deficit* over a directed link $(i, j) \in E$ by $Q_{ij}(t) = R_i(t) - R_j(t)$. Under a policy in Π^* , $Q_{ij}(t)$ is always nonnegative because, by part (1) of Lemma 2, we get

$$Q_{ij}(t) = R_i(t) - R_j(t) \geq \min_{k \in In(j)} R_k(t) - R_j(t) \geq 0.$$

The quantity $Q_{ij}(t)$ is the number of packets that have been received by a node i but not by node j . Intuitively, if all packet deficits $Q_{ij}(t)$ are bounded, then the total number of packets received by any node is not far from that generated at the source node r ; as a result, the broadcast throughput is equal to the packet arrival rate.

To analyze the system dynamics under a policy in Π^* , it is useful to define the *minimum packet deficit* at node j by

$$X_j(t) = \min_{i \in In(j)} Q_{ij}(t), \quad \forall j \neq r. \quad (10)$$

From part (2) of Lemma 2, $X_j(t)$ is the maximum number of packets that node j is allowed to receive from its in-neighbors in slot t . As an example, Fig. 4 shows that the packet deficits at node j , as compared to the upstream nodes a , b , and c , are $Q_{aj}(t) = 8$, $Q_{bj}(t) = 5$, and $Q_{cj}(t) = 4$, respectively. Thus

²If the network contains a directed cycle, then a deadlock may occur under a policy in Π^* and yields zero broadcast throughput. This problem does not arise when the network is a DAG.

$X_j(t) = 4$ and node j is only allowed to receive four packets in slot t due to Constraint 2. We can rewrite $X_j(t)$ as

$$X_j(t) = Q_{i_t^* j}(t), \quad \text{where } i_t^* = \arg \min_{i \in \text{In}(j)} Q_{ij}(t), \quad (11)$$

and the node i_t^* is the in-neighbor of node j from which node j has the smallest packet deficit in slot t ; ties are broken arbitrarily in deciding i_t^* .³ Our optimal broadcast policy will use the minimum packet deficits $X_j(t)$.

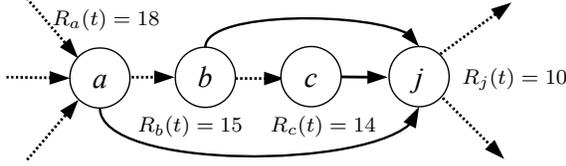


Fig. 4: Under a policy $\pi \in \Pi^*$, the set of packets available for transmission to node j in slot t is $\{11, 12, 13, 14\}$, which are available at all in-neighbors of node j . The in-neighbor of j inducing the smallest packet deficit is $i_t^* = c$, and $X_j(t) = \min\{Q_{aj}(t), Q_{bj}(t), Q_{cj}(t)\} = 4$.

B. The dynamics of the system variable $X_j(t)$

We analyze the dynamics of the system variables

$$X_j(t) = Q_{i_t^* j}(t) = R_{i_t^*}(t) - R_j(t), \quad (12)$$

under a policy $\pi \in \Pi^*$. Define the service rate vector $\mu(t) = (\mu_{ij}(t))_{i,j \in V}$ by

$$\mu_{ij}(t) = \begin{cases} c_{ij} & \text{if } (i, j) \in E \text{ and the link } (i, j) \text{ is activated,} \\ 0 & \text{otherwise.} \end{cases}$$

Equivalently, we may write $\mu_{ij}(t) = c_{ij}s_{ij}(t)$, and the number of packets forwarded over a link is constrained by the choice of the link-activation vector $s(t)$. At node j , the increase in the value of $R_j(t)$ depends on the identity of the received packets; in particular, node j must receive distinct packets. Next, we clarify which packets are to be received by node j .

The number of available packets for reception at node j is $\min\{X_j(t), \sum_{k \in V} \mu_{kj}(t)\}$, because: (i) $X_j(t)$ is the maximum number of packets node j can receive from its in-neighbors due to Constraint 2; (ii) $\sum_{k \in V} \mu_{kj}(t)$ is the total incoming transmission rate at node j under a given link-activation decision. To correctly derive the dynamics of $R_j(t)$, we consider the following efficiency requirement on policies in Π^* :

Constraint 3 (Efficient forwarding). *Given a service rate vector $\mu(t)$, node j pulls from the activated incoming links the following subset of packets*

$$\left\{ p \mid R_j(t) + 1 \leq p \leq R_j(t) + \min\left\{X_j(t), \sum_{k \in V} \mu_{kj}(t)\right\} \right\}, \quad (13)$$

³We note that the minimizer i_t^* is a function of the node j and the time slot t ; we slightly abuse the notation by neglecting j to avoid clutter.

where which packets are pulled over each incoming link are arbitrary but must satisfy Constraint 1.⁴

Constraint 3 states that scheduling policies must avoid forwarding the same packet to a node over two different incoming links in a slot. Under certain interference models such as the primary interference model, at most one incoming link is activated at a node in a slot, and Constraint 3 is redundant.

In (12), the packet deficit $Q_{i_t^* j}(t)$ increases with $R_{i_t^*}(t)$ and decreases as $R_j(t)$ increases, where $R_{i_t^*}(t)$ and $R_j(t)$ are both non-decreasing. It follows that we can upper bound the increase of $Q_{i_t^* j}(t)$ by the total capacity $\sum_{m \in V} \mu_{mi_t^*}(t)$ of activated incoming links at node i_t^* . Also, we can express the decrease of $Q_{i_t^* j}(t)$ by the exact number of distinct packets received by node j from its in-neighbors, and it is given by $\min\{X_j(t), \sum_{k \in V} \mu_{kj}(t)\}$ by Constraint 3. Consequently, the one-slot evolution of the variable $Q_{i_t^* j}(t)$ is given by⁵

$$\begin{aligned} Q_{i_t^* j}(t+1) &\leq (Q_{i_t^* j}(t) - \sum_{k \in V} \mu_{kj}(t))^+ + \sum_{m \in V} \mu_{mi_t^*}(t) \\ &\leq (X_j(t) - \sum_{k \in V} \mu_{kj}(t))^+ + \sum_{m \in V} \mu_{mi_t^*}(t), \end{aligned} \quad (14)$$

where $(x)^+ = \max(x, 0)$ and we recall that $X_j(t) = Q_{i_t^* j}(t)$. It follows that $X_j(t)$ evolves over slot t according to

$$\begin{aligned} X_j(t+1) &\stackrel{(a)}{=} \min_{i \in \text{In}(j)} Q_{ij}(t+1) \stackrel{(b)}{\leq} Q_{i_t^* j}(t+1) \\ &\stackrel{(c)}{\leq} (X_j(t) - \sum_{k \in V} \mu_{kj}(t))^+ + \sum_{m \in V} \mu_{mi_t^*}(t), \end{aligned} \quad (15)$$

where (a) follows the definition of $X_j(t)$, (b) follows because node $i_t^* \in \text{In}(j)$, and (c) follows from (14). In (15), we abuse the notation to define $\sum_{m \in V} \mu_{mr}(t) = A(t)$ for the source node r , where $A(t)$ is the number of exogenous packet arrivals in slot t .

C. The optimal broadcast policy

Our broadcast policy is designed to keep the minimum deficits $X_j(t)$ bounded. For this, we regard the variables $X_j(t)$ as virtual queues that follow the dynamics (15). By performing drift analysis on the virtual queues $X_j(t)$, we propose the following max-weight-type broadcast policy that belongs to the policy subclass Π^* which enforces Constraints 1, 2, and 3. We will show that this policy achieves the broadcast capacity λ^* of a wireless network over the general policy class Π when the underlying network graph is a DAG.

Optimal Broadcast Policy π^* over a Wireless DAG:

In slot t , the algorithm has the input $\{R_j(t), j \in V\}$ and performs the following four steps.

⁴Due to Constraints 1 and 2, the packets in (13) have been received by all in-neighbors of node j .

⁵We emphasize that the node i_t^* is defined in (11), depends on the particular node j and time t , and may be different from the node i_{t+1}^* .

Step 1: For each link $(i, j) \in E$, compute the deficit $Q_{ij}(t) = R_i(t) - R_j(t)$ and the set of nodes $K_j(t)$ for which node j is the deficit minimizer:

$$K_j(t) = \{k \in V \mid j = \arg \min_{m \in \text{In}(k)} Q_{mk}(t)\}. \quad (16)$$

We note that the ties are broken arbitrarily in finding a deficit minimizer.

Step 2: Compute $X_j(t) = \min_{i \in \text{In}(j)} Q_{ij}(t)$ for $j \neq r$ and assign to link (i, j) the weight

$$W_{ij}(t) = (X_j(t) - \sum_{k \in K_j(t)} X_k(t))^+, \quad (17)$$

where $W_{ij}(t)$ is the minimum deficit of node j minus that of all nodes for which node j is the deficit minimizer. Intuitively, the term $W_{ij}(t)$ arises because while delivering a packet to node j decreases $X_j(t)$ by one, it also increases $X_k(t)$ by one for all nodes for which node j is an in-neighbor and the deficit minimizer.

Step 3: In slot t , choose the link-activation vector $\mathbf{s}(t) = (s_e(t), e \in E)$ such that

$$\mathbf{s}(t) \in \arg \max_{(s_e, e \in E) \in \mathcal{S}} \sum_{e \in E} c_e s_e W_e(t). \quad (18)$$

Every node $j \neq r$ uses activated incoming links to pull packets $\{R_j(t) + 1, \dots, R_j(t) + \min\{\sum_i c_{ij} s_{ij}(t), X_j(t)\}\}$ from its in-neighbors according to Constraint 3.

Step 4: The vector $(R_j(t), j \in V)$ is updated as follows:

$$R_j(t+1) = \begin{cases} R_j(t) + A(t), & j = r, \\ R_j(t) + \min\{\sum_i c_{ij} s_{ij}(t), X_j(t)\}, & j \neq r, \end{cases}$$

and $R_j(0) = 0$ for all $j \in V$.

We illustrate the above algorithm in an example in Fig. 5. The next theorem demonstrates the optimality of the broadcast policy π^* .

Theorem 2. *If the underlying network graph \mathcal{G} is a DAG, then for any exogenous packet arrival rate $\lambda < \lambda_{\text{DAG}}$, the broadcast policy π^* yields*

$$\lim_{T \rightarrow \infty} \frac{R_i^{\pi^*}(T)}{T} = \lambda, \quad \forall i \in V, \quad \text{with probability 1,}$$

where λ_{DAG} is the upper bound on the broadcast capacity λ^* in the general policy class Π , as shown in (8). Consequently, the broadcast policy π^* achieves the broadcast capacity λ^* when the network topology is a DAG.

Proof: See the technical report [13]. ■

D. Number of disjoint spanning trees in a DAG

Theorem 2 provides an interesting combinatorial result that relates the number of disjoint spanning trees in a DAG to the in-degrees of its nodes.

Lemma 3. *Consider a directed acyclic graph $G = (V, E)$ that is rooted at a node r , has unit-capacity links, and possibly*

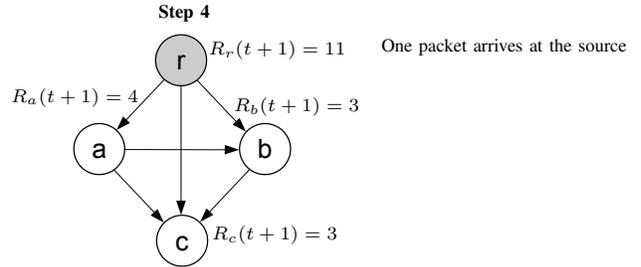
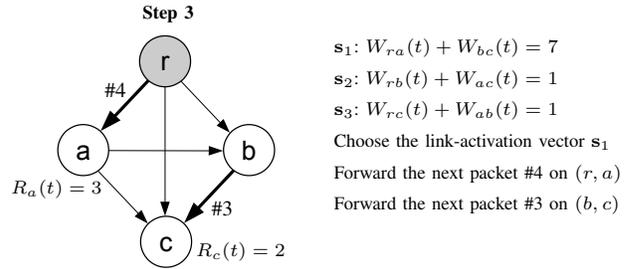
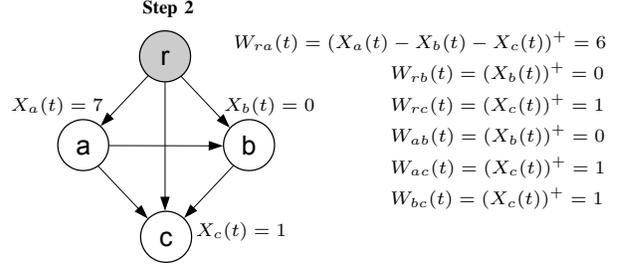
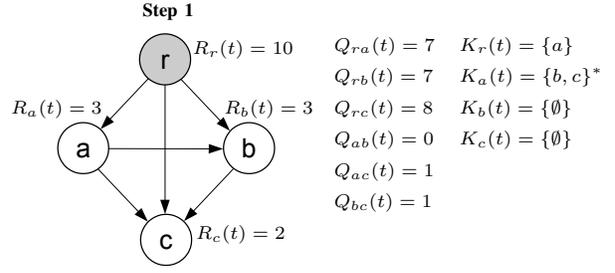


Fig. 5: Running the optimal broadcast policy π^* in slot t in a wireless network with unit-capacity links and under the primary interference constraint. Step 1: computing the deficits $Q_{ij}(t)$ and $K_j(t)$; a tie is broken in choosing node a as the in-neighbor deficit minimizer for node c , hence $c \in K_a(t)$; node b is also a deficit minimizer for node c . Step 2: computing $X_j(t)$ for $j \neq r$ and $W_{ij}(t)$. Step 3: finding the link activation vector that is a maximizer in (18), and forwarding the next in-order packets over activated links. Step 4: a new packet arrives at the source node r , and new values of $\{R_r(t+1), R_a(t+1), R_b(t+1), R_c(t+1)\}$ are updated.

contains parallel edges. The maximum number k^* of disjoint spanning trees in G is given by

$$k^* = \min_{v \in V \setminus \{r\}} d_{in}(v)$$

where $d_{in}(v)$ denotes the in-degree of the node v .

Proof of Lemma 3: See Appendix B. ■

V. SIMULATION RESULTS

We simulate the optimal broadcast policy π^* in a wireless DAG network with the primary interference constraint in Fig. 6; the link capacities are presented as weights on the links. The broadcast capacity λ^* of the network is upper bounded by the maximum throughput of node c , which is 1 because at most one of its incoming links can be activated at any time. To show that the broadcast capacity is indeed $\lambda^* = 1$, we consider the three spanning trees $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ rooted at the source node r . By finding the optimal time-sharing of all feasible link activations over a subset of spanning trees using linear programming, we can show that the maximum broadcast throughput using only the spanning tree \mathcal{T}_1 is $3/4$. The maximum broadcast throughput over the two trees $\{\mathcal{T}_1, \mathcal{T}_2\}$ is $6/7$, and that over all three trees $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ is 1. Thus, the upper bound is achieved and the broadcast capacity is $\lambda^* = 1$.

We compare our broadcast policy π^* with the tree-based policy π_{tree} in [4]. While the policy π_{tree} is originally proposed to transmit multicast traffic in a wired network by balancing traffic over multiple trees, we slightly modify the policy π_{tree} for broadcasting packets over spanning trees in the wireless setting; link activations are chosen according to the max-weight procedure. See Fig. 7 for a comparison of the average delay performance under the policy π^* and the tree-based policy π_{tree} over different subset of trees. The simulation duration is 10^5 slots. We observe that the policy π^* achieves the broadcast capacity $\lambda^* = 1$ and is throughput optimal.

The broadcast policy π^* does not rely on the limited tree structures and therefore has the potential to exploit all degrees of freedom in packet forwarding in the network; such freedom may lead to better delay performance as compared to the tree-based policy. To observe this effect, we consider the 10-node DAG network subject to the primary interference constraint in Fig. 8. For every pair of node $\{i, j\}$, $1 \leq i < j \leq 10$, the network has a directed link from i to j with capacity $(10 - i)$. By induction, we can calculate the number of spanning trees rooted at the source node 1 to be $9! \approx 3.6 \times 10^5$. We choose five arbitrary spanning trees $\{\mathcal{T}_i, 1 \leq i \leq 5\}$, over which the tree-based algorithm π_{tree} is simulated. Table I demonstrates the superior delay performance of the broadcast policy π^* , as compared to that of the tree-based algorithm π_{tree} over different subsets of the spanning trees. It also shows that a tree-based algorithm that does not use enough trees would result in degraded throughput.

VI. CONCLUSION

We characterize the broadcast capacity of a wireless network under general interference constraints. When the underlying network topology is a DAG, we propose a dynamic algorithm that achieves the wireless broadcast capacity. Our novel design, based on packet deficits and the in-order packet delivery constraint, is promising for application to other systems with packet replicas, such as multicasting and caching systems.

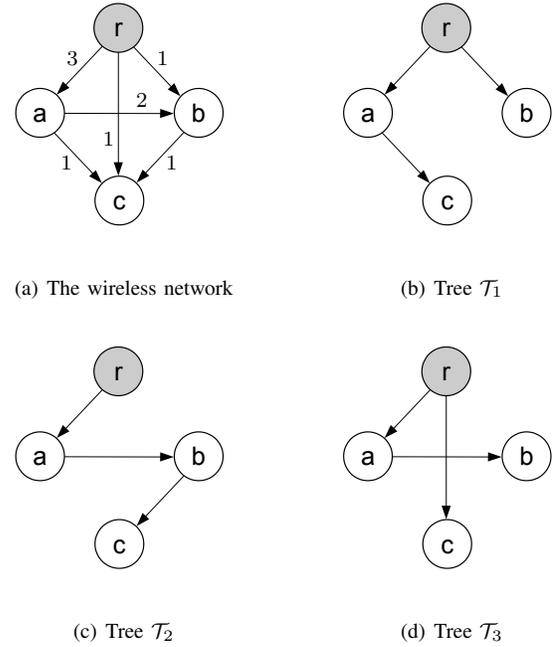


Fig. 6: A wireless DAG network and its three embedded spanning trees.

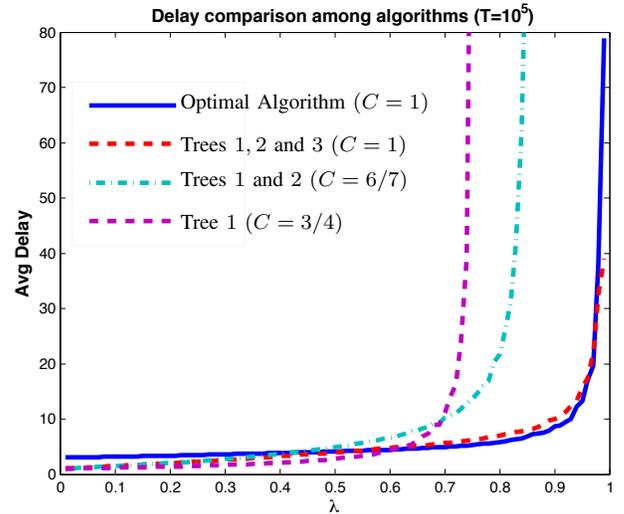


Fig. 7: Average delay performance of the optimal broadcast policy π^* and the tree-based policy π_{tree} that balances traffic over different subsets of spanning trees.

λ	tree-based policy π_{tree} over the spanning trees:					broadcast policy π^*
	\mathcal{T}_1	$\mathcal{T}_1 \sim \mathcal{T}_2$	$\mathcal{T}_1 \sim \mathcal{T}_3$	$\mathcal{T}_1 \sim \mathcal{T}_4$	$\mathcal{T}_1 \sim \mathcal{T}_5$	
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 I: Average delay performance of the tree-based policy π_{tree} over different subsets of spanning trees and the optimal broadcast policy π^* .

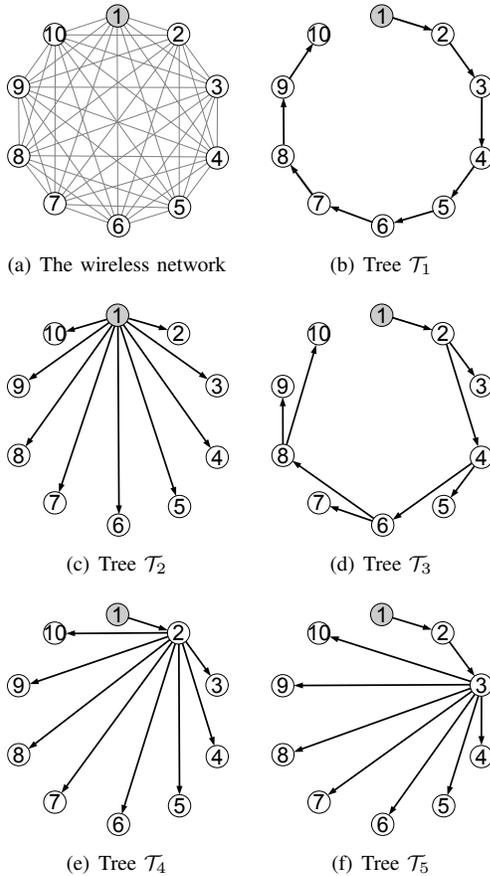


Fig. 8: The 10-node wireless DAG network and a subset of spanning trees.

Future work involves the study of cyclic networks, where optimal policies must be sought in the class $\Pi \setminus \Pi^{\text{in-order}}$.

REFERENCES

- [1] A. Karam, L. Zhang, and A. Lakas, "An efficient broadcasting scheme in support of military ad hoc communications in battle field," in *Innovations in Information Technology (IIT), 2013 9th International Conference on*. IEEE, 2013, pp. 78–83.
- [2] Livestream[®]. [Online]. Available: <http://new.livestream.com/>
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, Aug 2002.

- [4] S. Sarkar and L. Tassiulas, "A framework for routing and congestion control for multicast information flows," *Information Theory, IEEE Transactions on*, vol. 48, no. 10, pp. 2690–2708, 2002.
- [5] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *Automatic Control, IEEE Transactions on*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [6] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [7] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 2. IEEE, 2003, pp. 1124–1130.
- [8] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless networks*, vol. 8, no. 2-3, pp. 153–167, 2002.
- [9] L. Massoulié, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 1073–1081.
- [10] D. Towsley and A. Twigg, "Rate-optimal decentralized broadcasting: the wireless case," in *ACITA*, 2008.
- [11] S. Zhang, M. Chen, Z. Li, and L. Huang, "Optimal distributed broadcasting with per-neighbor queues in acyclic overlay networks with arbitrary underlay capacity constraints," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 814–818.
- [12] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *In Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, 2005.
- [13] "Throughput-optimal broadcast on directed acyclic graphs," <http://arxiv.org/abs/1411.6172>, Tech. Rep.
- [14] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.
- [15] R. Rustin, *Combinatorial Algorithms*. Algorithmics Press, 1973.

APPENDIX

A. Proof of Lemma 1

Consider the cyclic wired network in Fig. 9(a), where all edges have unit capacity and there is no interference constraint. Node a has total incoming capacity equal to two; thus, the broadcast capacity of the network is upper bounded by $\lambda^* \leq 2$. In fact, the network has two edge-disjoint spanning trees as shown in Figures 9(b) and 9(c). We can achieve the broadcast capacity $\lambda^* = 2$ by routing odd and even packets along the trees \mathcal{T}_1 and \mathcal{T}_2 , respectively.

Consider a policy $\pi \in \Pi_{\text{in-order}}$ that provides in-order delivery of packets to all network nodes. Let $R_i(t)$ be the number of distinct packets received by node i up to time t ; node i receives packets $\{1, 2, \dots, R_i(t)\}$ by time t due to in-order packet delivery. Consider the directed cycle $a \rightarrow b \rightarrow c \rightarrow a$

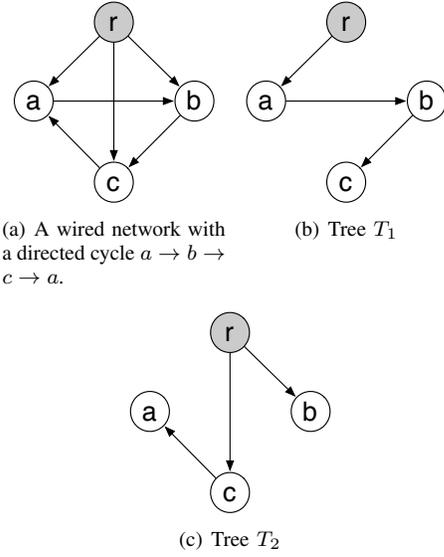


Fig. 9: A cyclic wired network and its two edge-disjoint spanning trees that yield the broadcast capacity $\lambda^* = 2$.

in Fig. 9(a). The necessary condition for all links in the cycle to forward packets in slot t is $R_a(t) > R_b(t) > R_c(t) > R_a(t)$, which is infeasible. Thus, there must exist an idle link in the cycle in every slot. Define the indicator variable $x_e(t) = 1$ if link e is idle in slot t under policy π , and $x_e(t) = 0$ otherwise. Since at least one link in the cycle is idle in every slot, we have

$$x_{(a,b)}(t) + x_{(b,c)}(t) + x_{(c,a)}(t) \geq 1.$$

Taking a time average of the above inequality yields

$$\frac{1}{T} \sum_{t=1}^T (x_{(a,b)}(t) + x_{(b,c)}(t) + x_{(c,a)}(t)) \geq 1.$$

Taking a lim sup at both sides, we obtain

$$\begin{aligned} & \sum_{e \in \{(a,b), (b,c), (c,a)\}} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_e(t) \\ & \geq \limsup_{T \rightarrow \infty} \sum_{e \in \{(a,b), (b,c), (c,a)\}} \frac{1}{T} \sum_{t=1}^T x_e(t) \\ & \geq 1. \end{aligned}$$

The above inequality implies that

$$\max_{e \in \{(a,b), (b,c), (c,a)\}} \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_e(t) \geq \frac{1}{3}. \quad (19)$$

Since the nodes $\{a, b, c\}$ are symmetrically located (i.e., the graph obtained by permuting the nodes $\{a, b, c\}$ is isomorphic to the original graph), without any loss of generality we may assume that the link $e = (a, b)$ attains the maximum in (19), i.e.,

$$\limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{(a,b)}(t) \geq \frac{1}{3}. \quad (20)$$

Noting that $x_e(t) = 1$ if link e is idle in slot t and that node b receives packets from nodes r and a , we can upper bound $R_b(T)$ by

$$\begin{aligned} R_b(T) & \leq \sum_{t=1}^T (1 - x_{(r,b)}(t) + 1 - x_{(a,b)}(t)) \\ & \leq \sum_{t=1}^T (2 - x_{(a,b)}(t)). \end{aligned}$$

It follows that

$$\liminf_{T \rightarrow \infty} \frac{R_b(T)}{T} \leq 2 - \limsup_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{(a,b)}(t) \leq \frac{5}{3}$$

where the last inequality uses (20). Thus, we have

$$\min_{i \in V} \liminf_{T \rightarrow \infty} \frac{R_i(T)}{T} \leq \liminf_{T \rightarrow \infty} \frac{R_b(T)}{T} \leq \frac{5}{3},$$

which holds for all policies $\pi \in \Pi_{\text{in-order}}$. Taking the supremum over the policy class $\Pi_{\text{in-order}}$ shows that the broadcast capacity $\lambda_{\text{in-order}}^*$ subject to the in-order packet delivery constraint satisfies

$$\begin{aligned} \lambda_{\text{in-order}}^* & = \sup_{\pi \in \Pi_{\text{in-order}}} \min_{i \in V} \liminf_{T \rightarrow \infty} \frac{R_i(T)}{T} \\ & \leq \frac{5}{3} < 2 = \lambda^* \end{aligned}$$

Here, we use the more rigorous definition of a broadcast policy of rate λ in (2). Hence the network broadcast capacity is strictly reduced by in-order packet delivery in the cyclic network in Fig. 9(a).

B. Proof of Lemma 3

We regard the DAG G as a wired network in which all links can be activated simultaneously. Theorem 2 and (9) show that the broadcast capacity of the wired network G is

$$\lambda^* = \lambda_{\text{DAG}} = \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \quad (21)$$

$$= \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e = \min_{v \in V \setminus \{r\}} d_{\text{in}}(v) \quad (22)$$

where $U_v = V \setminus \{v\}$ is the proper cut that separates node v from the network, E_{U_v} is the set of incoming links of node v , and the last equality follows that the maximizer in (9) is the all-one vector $\beta = \mathbf{1}$ and that all links have unity capacity. The Edmond's theorem [15] states that the maximum number of disjoint spanning trees in the directed graph G is

$$k^* = \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e. \quad (23)$$

Combining (22) and (23) completes the proof.