Throughput-Optimal Multihop Broadcast on Directed Acyclic Wireless Networks

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: *sinhaa@mit.edu, †georgios.paschos@huawei.com, ‡cpli@qti.qualcomm.com, *modiano@mit.edu

Abstract—We study the problem of efficiently broadcasting packets in multi-hop wireless networks. At each time slot the network controller activates a set of non-interfering links and forwards selected copies of packets on each activated link. A packet is considered jointly received only when *all nodes* in the network have obtained a copy of it. The maximum rate of jointly received packets is referred to as the broadcast capacity of the 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 is decentralized, utilizes local queue-length information only and does not require the use of global topological structures such as spanning trees. The principal technical challenge inherent in the problem is the absence of work-conservation principle due to the duplication of packets, which renders traditional queuing modelling inapplicable. We overcome this difficulty by studying relative packet deficits and imposing in-order delivery constraints to every node in the network. Although in-order packet delivery, in general, leads to degraded throughput in graphs containing cycles, we show that it is throughput optimal in DAGs and can be exploited to simplify the design and analysis of optimal algorithms. Our characterization leads to a polynomial time algorithm for computing the broadcast capacity of any wireless DAG under the primary interference constraints. Additionally, we propose a multiclass extension of our algorithm which can be effectively used for broadcasting in any network with arbitrary topology. Simulation results show that the our algorithm has superior delay performance as compared to the tree-based approaches.

I. INTRODUCTION AND RELATED WORK

Broadcast refers to the fundamental network functionality of delivering data from a source node to all other nodes. For efficient broadcasting, we need to use appropriate packet replication and 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 every time slot. 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. These algorithms, however, are not suitable for wireless networks because enumerating all spanning trees is computationally prohibitive that needs to be performed repeatedly when the network topology changes with time.

In this paper, we study the fundamental problem of throughput optimal broadcasting in wireless networks. We consider a time-slotted system. At every slot, a scheduler decides which non-interfering wireless links to activate and which set of packets to forward over the 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. To the best of our knowledge, there does not exist any capacity-achieving scheduling policy for wireless broadcast without the use of spanning trees [1]. The main contribution of this paper is to design provably optimal wireless broadcast algorithms that does not use spanning trees when the underlying topology is a DAG.

We start out with considering a rich class of scheduling policies Π that perform arbitrary link activations and packet forwarding. We define the broadcast capacity λ^* as the maximum common rate achievable over this policy class Π. We next enforce two constraints that lead to a smaller set of policies. 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 neighbours 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

[1]Note that we exclude network-coding operations throughout the paper.

the broadcast capacity. In contrast, we prove the existence of a network containing a cycle in which no control policy in the policy-space $\Pi^{\text{in-order}}$ can achieve the broadcast capacity.

To enable the design of the optimal broadcast policy, we establish a *queue-like dynamics* for the system-state, represented by relative packet deficits. This is non-trivial for the broadcast problem because explicit queueing structure is difficult to define in the network due to packet replications. We subsequently show that, the problem of achieving the broadcast capacity reduces to finding a scheduling policy *stabilizing* the system, which can be accomplished by stochastic Lyapunov drift analysis techniques [5], [6].

In this paper, we make the following contributions:

- 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. This algorithm also yields a constructive proof of a version of Edmonds' disjoint tree-packing theorem [7] which is generalized to wireless activations but specialized to DAG topology.

- Based on our characterization of the broadcast capacity, we derive a polynomial-time algorithm to compute the broadcast capacity of any wireless DAG under primary interference constraints.

- We propose a randomized multiclass extension of our algorithm, which can be effectively used to do broadcast on wireless networks with arbitrary underlying topology.

- We demonstrate the superior delay performance of our DAG-policy, as compared to centralized tree-based algorithm [4], via numerical simulations. We also explore the efficiency/complexity trade-off of our proposed multiclass extension through extensive simulations.

In the literature, a simple method for wireless broadcast is to use packet flooding [8]. The flooding approach, however, leads to redundant transmissions and collisions, known as *broadcast storm* [9]. In the wired domain, it has been shown that forwarding *useful* packets at random is optimal for broadcast [10]; this approach does not extend to the wireless setting due to interference and the need for scheduling [11]. Broadcast on wired networks can also be done using network coding [12], [13]. 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 cut-set consideration. In Section IV, we propose a dynamic broadcast policy that achieves the broadcast capacity in a DAG. In section V, we propose an efficient algorithm for computing the broadcast capacity

of any wireless DAG under primary interference constraints. Our DAG-broadcast algorithm is extended to networks with arbitrary topology in section VI. Illustrative simulation results are presented in Section VII. Finally, we conclude our paper in section VIII.

II. THE WIRELESS NETWORK MODEL

We consider a time-slotted wireless network model represented by the tuple $\left(\mathcal{G}(V,E), \boldsymbol{c}, \mathcal{S}\right)$, where V is the set of nodes, E is the set of directed links, $\boldsymbol{c} = (c_e, e \in E)$ is the capacity-vector of the links and $\mathcal{S}$ is the set of all feasible link-activation vectors, whose elements $\boldsymbol{s} = (s_e, e \in E) \in \mathcal{S}$ are binary vectors such that the links $e \in E$ with $s_e = 1$ can be activated simultaneously. The structure of the activation set $\mathcal{S}$ depends on the underlying interference model. For example, under the *primary* interference constraint (also known as *node-exclusive interference constraint* [14]), the set $\mathcal{S}$ consists of all binary vectors corresponding to matchings of the underlying graph $\mathcal{G}$ [15], 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 stochastic broadcast traffic is generated (or arrives externally). The number of packets generated at the node r at slot t is given by the random variable $A(t) \in \mathbb{Z}_+$, which is i.i.d. over slots with mean λ. These packets are to be delivered efficiently to all other nodes in the network.

(a) a wireless network

(b) activation vector $\boldsymbol{s}_1$

(c) activation vector $\boldsymbol{s}_2$

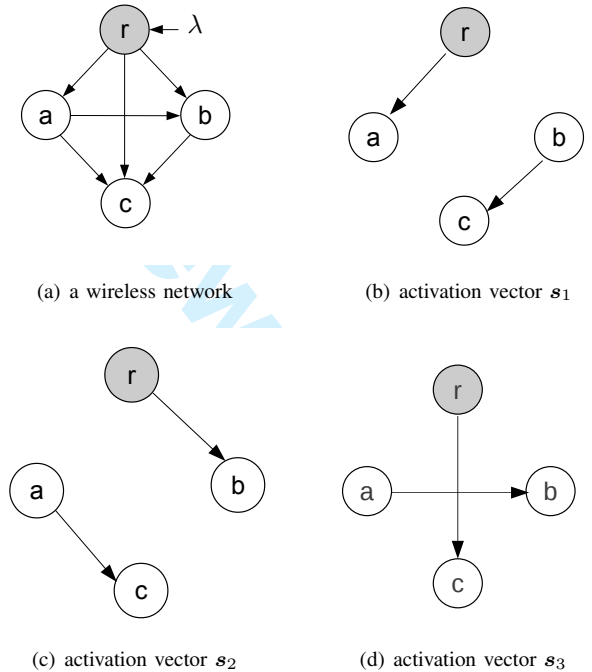(d) activation vector $\boldsymbol{s}_3$

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 $\pi \in \Pi$ consists of a sequence of actions $\{\pi_t\}_{t \geq 1}$ executed at every slot t. Each action π_t comprises of two operations: (i) the scheduler activates a subset of links by choosing a feasible activation vector $\boldsymbol{s}(t) \in \mathcal{S}$; (ii) each node i forwards a subset of packets (possibly empty) to node j over an activated link $(i, j) \in \mathbb{1}(\boldsymbol{s}(t) = 1)$, subject to the link capacity constraint. The class Π includes policies that use all past and future information, and may forward any subset of packets over a link.

Let $R_i^\pi(t)$ 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 $\liminf_{T \to \infty} R_i^\pi(T)/T$ 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.,*

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

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

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 $\pi \in \Pi$ of rate λ.*

A. An upper bound on broadcast capacity λ^*

We characterize the broadcast capacity λ^* of a wireless network by proving 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-averaged behaviour of the wireless network. We first give an intuitive explanation of the bound, assuming that the involved limits exist. Then in the proof of Theorem 1 we rigorously prove the bound by relaxing this assumption.

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

$$\boldsymbol{\beta}^\pi = (\beta_e^\pi, e \in E) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t), \quad (2)$$

where $\boldsymbol{s}^\pi(t)$ is the link-activation vector under policy π in slot t. The average flow rate over a link e under the policy π is upper bounded by the product of the link capacity and the fraction of time the link e is activated, i.e., $c_e \beta_e^\pi$. Hence, we can define an edge-capacitated graph $\widehat{\mathcal{G}}^\pi = (V, E, (\widehat{c}_e))$ associated with policy π, where each directed link $e \in E$ has capacity $\widehat{c}_e = c_e \beta_e^\pi$; 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 capacity on the ensemble of graphs $\widehat{\mathcal{G}}^\pi$ over all feasible vectors $\boldsymbol{\beta}^\pi$.

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

$$E_U = \{(i, j) \in E \mid i \in U, \; j \notin U\}. \quad (3)$$

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
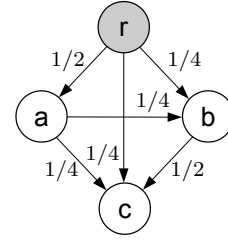

Fig. 2: The edge-capacitated graph $\widehat{\mathcal{G}}^\pi$ for the wireless network with unit link capacities in Fig. 1 and under the time-average vector $\boldsymbol{\beta}^\pi = (1/2, 1/4, 1/4)$. The link weights are the capacities $c_e \beta_e^\pi$. The minimum proper cut in this graph has value $1/2$ (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 $\boldsymbol{\beta}^\pi \in \text{conv}(\mathcal{S})$.

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 an upper-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

$$\lambda^\pi \leq \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e^\pi. \quad (4)$$

Equation (4) holds for any policy $\pi \in \Pi$. Thus, the broadcast capacity λ^* of the wireless network satisfies

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

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

where the last inequality holds because the vector $\boldsymbol{\beta}^\pi$ associated with any policy $\pi \in \Pi$ lies in the convex hull of the activation set $\mathcal{S}$. Our first theorem formalizes the above intuitive characterization of the broadcast capacity λ^* of a wireless network.

Theorem 1. *The broadcast capacity λ^* of a wireless network $\mathcal{G}(V, E, \boldsymbol{c})$ with activation set $\mathcal{S}$ is upper bounded as follows:*

$$\lambda^* \leq \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \left(\min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e \right). \quad (5)$$

Proof of Theorem 1: See Appendix A. ∎

B. In-order packet delivery

Studying the performance of any arbitrary broadcast policy $\pi \in \Pi$ is formidable because packets are replicated across the network and may be received out of order. To avoid unnecessary re-transmissions, the nodes must keep track of the identity of the received set of packets, which complicates the system state; because instead of the number of packets received, the system state is properly described here by the subset of packets received at each of the nodes.

To simplify the system state, we focus on the subset of policies $\Pi^{\text{in-order}} \subset \Pi$ that enforce the following constraint:

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

In-order packet delivery is useful in live media streaming applications [2], where 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 $\{1, \ldots, R_i(t)\}$. Therefore, the network state in slot t is given by the vector $\boldsymbol{R}(t) = \big(R_i(t), i \in V\big)$.

In section IV we will prove that there exists a throughput-optimal broadcast policy in the space $\Pi^{\text{in-order}}$ when the underlying network topology is a DAG. Ironically, Lemma (1) shows that there exists a network containing a cycle in which any broadcast policy in the space $\Pi^{\text{in-order}}$ is *not* throughput optimal. Hence the space $\Pi^{\text{in-order}}$ can not, in general, be extended beyond DAGs while preserving throughput optimality.

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 containing a directed cycle such that $\lambda^*_{\text{in-order}} < \lambda^*$.*

Proof of Lemma 1: See Appendix B. ∎

We will return to the problem of broadcasting in networks with arbitrary topology in Section VI.

C. Achieving the broadcast capacity in a DAG

At this point we concentrate our attention to Directed Acyclic Graphs (DAGs). Graphs in this class are appealing for our analysis because they possess well-known topological ordering of the nodes [15]. For DAGs, the upper bound (5) on the broadcast capacity λ^* in Theorem 1 will be simplified further. For each receiver node $v \neq r$, consider the proper cut U_v that separates the network from node v:

$$U_v = V \setminus \{v\}. \tag{6}$$

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

$$\lambda_{\text{DAG}} \triangleq \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e \, \beta_e \tag{7}$$

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

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 $\Pi_{\text{in-order}}$ and achieves the broadcast rate λ_{DAG}. Combining this result with (7), we establish that the broadcast capacity of a DAG is given by

$$\lambda^* = \lambda_{\text{DAG}} = \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \min_{\{U_v, v \neq r\}} \sum_{e \in E_{U_v}} c_e \, \beta_e, \tag{8}$$

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

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

From a computational point of view, the equality in Eqn. (8) is attractive, because it implies that for computing the broadcast capacity of any wireless DAG, it is enough to consider only those cuts that separate a single (non-source) node from the source-side. Note that, there are only $|V| - 1$ of such cuts, in contrast with the total number of cuts, which is exponential in the size of the network. This fact will be exploited in section V to develop a strongly poly-time algorithm for computing the broadcast capacity of any DAG under the primary interference constraints.

IV. DAG Broadcast Algorithm

In this section we design an optimal broadcast policy for wireless DAGs. We start by imposing an additional constraint that leads to a new subclass of policies $\Pi^* \subseteq \Pi^{\text{in-order}}$. As we will see, policies in Π^* can be described in terms of relative packet deficits which constitute a simple dynamics. We analyze the dynamics of the minimum relative packet deficit at each node j, where the minimization is over all incoming neighbours of j. This quantity plays the role of virtual queues in the system and we design a dynamic 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 showed in Section III-B that, constrained to the policy-space $\Pi^{\text{in-order}}$, the system-state is completely represented by the vector $\boldsymbol{R}(t)$. To simplify the system dynamics further, we restrict $\Pi^{\text{in-order}}$ further as follows.

We say that node i is an *in-neighbor* of node j iff 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 at a slot t only if all the in-neighbours of j have received packet p in some previous slot.*

We denote this new policy-class by $\Pi^* \subseteq \Pi^{\text{in-order}}$. We will soon show that it contains an optimal policy. Fig. 3 shows the relationship among different policy classes[2].

Following properties of the system-states $\boldsymbol{R}(t)$ under a policy $\pi \in \Pi^*$ will be useful.

Lemma 2. *For $j \neq r$, 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) $R_j(t) \leq \min_{i \in In(j)} R_i(t)$

(2) *The indices of packets p that are eligible to be transmitted to the node j at slot t is given by*

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

---

[2]We note that, if the network contains a directed cycle, then a deadlock might occur under a policy in Π^* and may yield zero broadcast throughput. However, this problem does not arise when the underlying topology is a DAG.

Π: all policies that perform link activations and routing

Π^{in-order}: policies that enforce in-order packet delivery

Π*: policies that allow reception only if all in-neighbors have received the specific packet
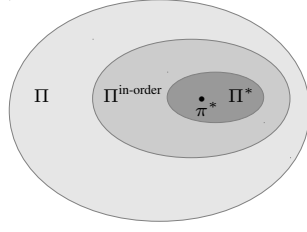
Fig. 3: Containment relationships among different policy classes.

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 non-negative because, by part (1) of Lemma 2, we have

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

The quantity $Q_{ij}(t)$ denotes the number of packets received by node i but not by node j, upto time t. Intuitively, if all packet deficits $Q_{ij}(t)$ are bounded asymptotically, the total number of packets received by any node is not lagging far from the total number of packets generated at the source; hence, the broadcast throughput will be equal to the packet generation rate.

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

$$X_j(t) = \min_{i \in \text{In}(j)} Q_{ij}(t). \tag{9}$$

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 at slot t. As an example, Fig. 4 shows that the packet deficits at node j, relative to the upstream nodes a, b, and c, are $Q_{aj}(t) = 8$, $Q_{bj}(t) = 5$, and $Q_{cj}(t) = 4$, respectively. Thus $X_j(t) = 4$ and node j is only allowed to receive four packets in slot t due to Constraint 2. We can rewrite $X_j(t)$ as

$$X_j(t) = Q_{i_t^* j}(t), \quad \text{where } i_t^* = \arg\min_{i \in \text{In}(j)} Q_{ij}(t), \tag{10}$$

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 be described in terms of the minimum packet deficits $\{X_j(t)\}$.
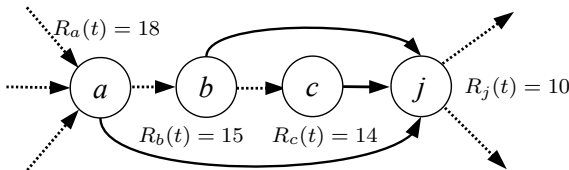

Fig. 4: Under a policy $\pi \in \Pi^*$, the set of packets available for transmission to node j in slot t is $\{11, 12, 13, 14\}$, which are available at all in-neighbors of node j. The in-neighbor of j inducing the smallest packet deficit is $i_t^* = c$, and $X_j(t) = \min\{Q_{aj}(t), Q_{bj}(t), Q_{cj}(t)\}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

---

[3]We 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.

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

We now analyze the dynamics of the system variables

$$X_j(t) = Q_{i_t^* j}(t) = R_{i_t^*}(t) - R_j(t) \tag{11}$$

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

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-neighbours subject to the 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 $\boldsymbol{\mu}(t)$, node j pulls from the activated incoming links the following subset of packets (denoted by their indices)*

$$\left\{ p \mid R_j(t) + 1 \leq p \leq R_j(t) + \min\{X_j(t), \sum_{k \in V} \mu_{kj}(t)\} \right\}, \tag{12}$$

The specific subset of packets that are pulled over each incoming link are disjoint but otherwise arbitrary.[4]

Constraint 3 requires that scheduling policies must avoid forwarding the same packet to a node over two different incoming links. 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 Eqn. (11), the packet deficit $Q_{i_t^* j}(t)$ increases with $R_{i_t^*}(t)$ and decreases with $R_j(t)$, where $R_{i_t^*}(t)$ and $R_j(t)$ are both non-decreasing. Hence, we can upper-bound the increment of $Q_{i_t^* j}(t)$ by the total capacity $\sum_{m \in V} \mu_{m i_t^*}(t)$ of the activated incoming links at node i_t^*. Also, we can express the decrement of $Q_{i_t^* j}(t)$ by the exact number of distinct packets received by node j from its in-neighbours, 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 by[5]

$$\begin{aligned} Q_{i_t^* j}(t+1) &\leq \left(Q_{i_t^* j}(t) - \sum_{k \in V} \mu_{kj}(t) \right)^+ + \sum_{m \in V} \mu_{m i_t^*}(t) \\ &= \left(X_j(t) - \sum_{k \in V} \mu_{kj}(t) \right)^+ + \sum_{m \in V} \mu_{m i_t^*}(t), \end{aligned} \tag{13}$$

---

[4]Due to Constraints 1 and 2, the packets in (12) have been received by all in-neighbors of node j.

[5]We emphasize that the node i_t^* is defined in (10), depends on the particular node j and time t, and may be different from the node i_{t+1}^*.

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

$$X_j(t+1) \stackrel{(a)}{=} \min_{i \in \text{In}(j)} Q_{ij}(t+1) \stackrel{(b)}{\leq} Q_{i_t^* j}(t+1)$$

$$\stackrel{(c)}{\leq} \big(X_j(t) - \sum_{k \in V} \mu_{kj}(t)\big)^+ + \sum_{m \in V} \mu_{mi_t^*}(t), \quad (14)$$

where the equality (a) follows the definition of $X_j(t)$, equality (b) follows because node $i_t^* \in \text{In}(j)$ and equality (c) follows from Eqn. (13). In Eqn. (14), if $i_t^* = r$, 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 generated at slot t.

C. The optimal broadcast policy

Our broadcast policy is designed to keep the minimum deficit process $\boldsymbol{X}(t)$ stable. For this, we regard the variables $X_j(t)$ as virtual queues that follow the dynamics (14). By performing drift analysis on the virtual queues $X_j(t)$, we propose the following max-weight-type broadcast policy π^*, described in Algorithm 1. We have $\pi^* \in \Pi^*$ and it enforces the 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 topology is a DAG.

Algorithm 1 Optimal Broadcast Policy π^* for a Wireless DAG:

At each slot t, the network-controller observes the state-variables $\{R_j(t), j \in V\}$ and executes the following actions

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 $K_j(t) \subset \text{out}(j)$ for which node j is their deficit minimizer, given as follows

$$K_j(t) \leftarrow \big\{k \in V \mid j = \arg \min_{m \in \text{In}(k)} Q_{mk}(t)\big\}. \quad (15)$$

The ties are broken arbitrarily (e.g., in favor of the highest indexed node) in finding the $\arg\min(\cdot)$ in Eqn.(15).

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

$$W_{ij}(t) \leftarrow \big(X_j(t) - \sum_{k \in K_j(t)} X_k(t)\big)^+. \quad (16)$$

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

$$\boldsymbol{s}(t) \in \arg \max_{\boldsymbol{s} \in \mathcal{S}} \sum_{e \in E} c_e s_e W_e(t). \quad (17)$$

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

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

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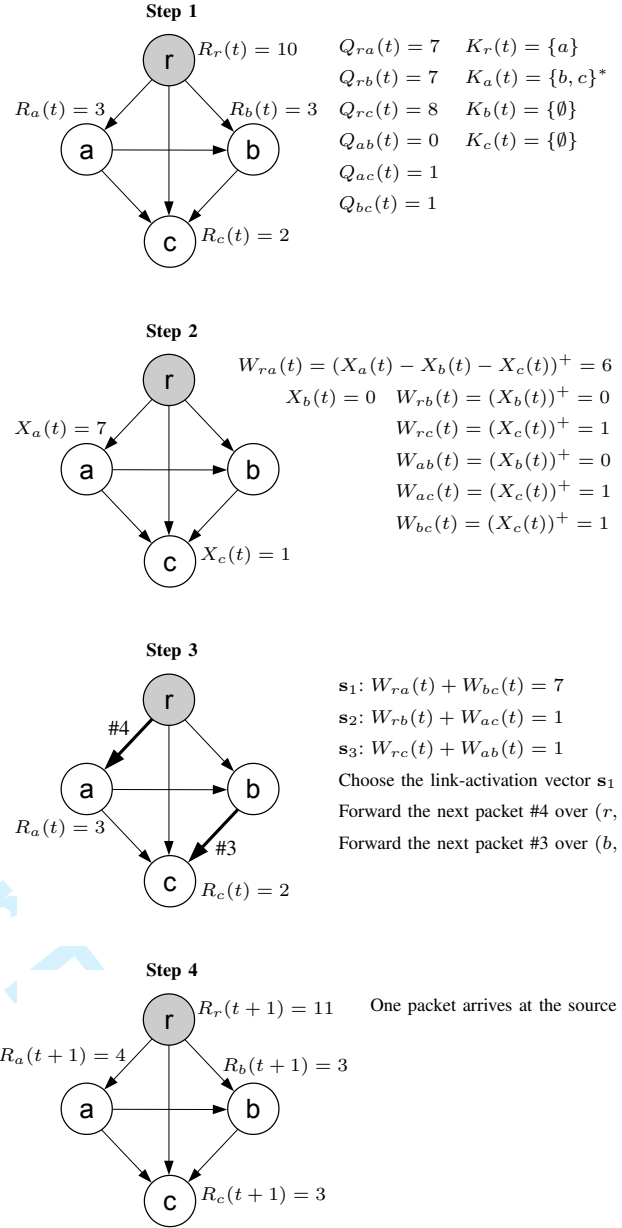

Fig. 5: Running the optimal broadcast policy π^* in slot t in a wireless network with unit-capacity links and under the primary interference constraint. Step 1: computing the deficits $Q_{ij}(t)$ and $K_j(t)$; a tie is broken in choosing node a as the in-neighbor deficit minimizer for node c, hence $c \in K_a(t)$; 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 (17) and forwarding the next in-order packets over the activated links. Step 4: a new packet arrives at the source node r and the values of $\{R_r(t+1), R_a(t+1), R_b(t+1), R_c(t+1)\}$ are updated.

We illustrate the Algorithm 1 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 $\lambda < \lambda_{DAG}$, the broadcast policy π^* yields*

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

where λ_{DAG} is the upper bound on the broadcast capacity λ^* in the general policy class Π, as shown in (7). Consequently, the broadcast policy π^* achieves the broadcast capacity λ^* for any Directed Acyclic Graphs.

Proof of Theorem 2: See Appendix C. ∎

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 contains parallel edges. The maximum number k^* of edge-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 D. ∎

V. EFFICIENT ALGORITHM FOR COMPUTING THE BROADCAST CAPACITY OF A DAG

In this section we exploit Eqn. (8) and develop an LP to compute the broadcast capacity of any wireless DAG network under the primary interference constraints. Although this LP has exponentially many constraints, using a well-known separation oracle, it can be solved in strongly polynomial time via the ellipsoid algorithm [16].

Under the primary interference constraint, the set of feasible activations of the graphs are *matchings* [15]. For a subset of edges $E' \subset E$, let $\chi^{E'} \in \{0,1\}^{|E|}$ where $\chi^{E'}(e) = 1$ if $e \in E'$ and is zero otherwise. Let us define

$$\mathcal{P}_{\text{matching}}(\mathcal{G}) = \textbf{convexhull}(\{\chi^M | M \text{ is a matching in } G\}) \quad (18)$$

We have the following classical result by Edmonds [17].

Theorem 3. *The set $\mathcal{P}_{matching}(\mathcal{G})$ is characterized by the set of all $\boldsymbol{\beta} \in \mathbb{R}^{|E|}$ such that :*

$$\beta_e \geq 0 \quad \forall e \in E \quad (19)$$

$$\sum_{e \in \delta_{in}(v) \cup \delta_{out}(v)} \beta_e \leq 1 \quad \forall v \in V$$

$$\sum_{e \in E[U]} \beta_e \leq \frac{|U| - 1}{2}; \quad U \subset V, |U| \text{ odd}$$

Here $E[U]$ is the set of edge (ignoring their directions) with both end points in U, $\delta_{in}(u)$ ($\delta_{out}(u)$) denotes the set of all incoming (outgoing) edges to (from) the vertex $u \in V$.

Hence following Eqn. (8), the broadcast capacity of a DAG can be obtained by the following LP :

$$\max \lambda \quad (20)$$

Subject to,

$$\lambda \leq \sum_{e \in \delta_{\text{in}}(v)} c_e \beta_e \quad \forall v \in V \setminus \{r\} \quad (21)$$

$$\boldsymbol{\beta} \in \mathcal{P}_{\text{matching}}(\mathcal{G}) \quad (22)$$

From the equivalence of optimization and separation (via the ellipsoid method), it follows that the above LP is poly-time solvable if there exists an efficient separator oracle for the constraints (21), (22). Since there are only linearly many constraints ($|V| - 1$, to be precise) in (21), the above requirement reduces to an efficient separator for the matching polytope (22). We refer to a classic result from the combinatorial-optimization literature which shows the existence of such efficient separator for the matching polytope

Theorem 4. *[17] There exists a strongly poly-time algorithm, that given $\mathcal{G} = (V, E)$ and $\boldsymbol{\beta} : E \to \mathbb{R}^{|E|}$ determines if $\boldsymbol{\beta}$ satisfies (19) or outputs an inequality from (19) that is violated by $\boldsymbol{\beta}$.*

This directly leads to the following theorem.

Theorem 5. *There exists a strongly poly-time algorithm to compute the broadcast capacity of any wireless DAG under the primary interference constraints.*

The following corollary implies that, although there are exponentially many matchings in a DAG, to achieve the broadcast capacity, randomly activating (with appropriate probabilities) only $|E| + 1$ matchings suffice.

Corollary 1. *The optimal broadcast capacity λ^* in a wireless DAG, under the primary interference constraints, can be achieved by randomly activating (with positive probability) at most $|E| + 1$ matchings.*

Proof: Let $(\lambda^*, \boldsymbol{\beta}^*)$ be an optimal solution of the LP (20). Hence we have $\boldsymbol{\beta}^* \in \mathcal{P}_{\text{matching}}(\mathcal{G}) \equiv$ **convexhull**($\{\chi^M | M \text{ is a matching in } G\}$). Since the polytope $\mathcal{P}_{\text{matching}}(\mathcal{G})$ is a subset of $\mathbb{R}^{|E|}$, by Carathéodory's theorem [18], the vector $\boldsymbol{\beta}^*$ can be expressed as a convex combination of at most $|E|+1$ vertices of the polytope $\mathcal{P}_{\text{matching}}(\mathcal{G})$, which are matchings of the graph $\mathcal{G}$. This concludes the proof. ∎

VI. BROADCASTING ON NETWORKS WITH ARBITRARY TOPOLOGY

In this section we extend the broadcast policy for a DAG to networks containing cycles. From the negative result of Lemma 1, we know that any policy ensuring *in-order* packet delivery at every node cannot, in general, achieve the broadcast capacity of a network containing cycles. To get around this difficulty, we introduce the concept of broadcasting using multiple *classes* $\mathcal{K}$ of packets. The idea is as follows: each class $k \in \mathcal{K}$ has a one-to-one correspondence with a specific

permutation $\prec_k$ of the nodes; for an edge $(a,b) \in E$ if the node a appears prior to the node b in the permutation $\prec_k$ (we denote this condition by $a \prec_k b$), then the edge (a,b) is included in the class k, otherwise the edge (a,b) ignored by the class k. The set of all edges included in the class k is denoted by $E^k \subset E$. It is clear that each class k corresponds to a unique embedded DAG topology $\mathcal{G}^k(V, E^k)$, which is a subgraph of the underlying graph $\mathcal{G}(V, E)$.

A new incoming packet arriving at the source node is admitted to some class $k \in \mathcal{K}$, according to some policy. All packets in a given class $k \in \mathcal{K}$ are broadcasted while maintaining in-order delivery property within the class k, however packets from different classes do not need to respect this constraint. Hence the resulting policy does not belong to the class Π^* in but rather to the general class Π. This new policy keeps the best of both worlds: (a) its description-complexity is $\Theta(kN)$, where for each class we essentially have the same representations as in the in-order delivery constrained policies and (b) by relaxing the inter-class in-order delivery constraint it has the potential to achieve the full broadcast capacity of the underlying graph. Hence the broadcast problem reduces to construction of multiple classes (which are permutations of the vertices V) out of the given directed graph such that it covers the graph efficiently, from a broadcast-capacity point of view. In Algorithm-2, we choose the permutations uniformly at random with the condition that the source always appears at the first position of the permutation.

Theorem 6. *The multiclass broadcast Algorithm-2 with K classes supports a broadcast rate of*

$$\lambda^K = \max_{\sum_k \boldsymbol{\beta}^k \in conv(\mathcal{S})} \sum_{k=1}^{K} \min_{j \neq r} \sum_i c_{ij} \beta_{ij}^k \qquad (26)$$

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

The right hand side of Eqn. (26) can be understood as follows. Consider a feasible stationary activation policy π_{STAT} which activates class l on the edge (i,j) β_{ij}^k fraction of time. Since, by construction, each of the class follows a DAG, lemma (3) implies that the resulting averaged graph has a broadcast capacity of $\lambda^k = \min_j \sum_i c_{ij} \beta_{ij}^k$ for the class k. Thus the total broadcast rate achievable by this scheme is simply $\lambda^K = \sum_{k=1}^{K} \lambda^k = \sum_k \min_j \sum_i c_{ij} \beta_{ij}^k$. Given the K classes, following the same line of argument as in (20), we can develop a similar LP to compute the broadcast capacity (26) of all these k-classes taken together in strongly poly-time.

The proof of Theorem (6) follows along the exact same line of argument as in Theorem (2), where we now work with the following Lyapunov function $\hat{L}(\boldsymbol{Q}(t))$, which takes into account all k classes:

$$\hat{L}(\boldsymbol{Q}(t)) = \sum_{k=1}^{K} \sum_{j \neq r} (X_j^k(t))^2 \qquad (27)$$

We then compare the multiclass broadcast algorithm 2 with the stationary activation policy π_{STAT} above to show that the Multiclass broadcast algorithm is stable under all arrival rates below λ. The details are omitted for brevity.

Algorithm 2 Multiclass Broadcast Algorithm for General Topology

Require: Graph $\mathcal{G}(V, E)$, total number of classes K
1: Generate K permutations $\{\prec_i\}_{i=1}^{K}$ of the nodes V uniformly at random (with the source $\{r\}$ at the first position) and obtain the induced DAGs $G^k(V, E^k)$, where $e = (a,b) \in E^k$ iff $a \prec_k b$.
2: For each permutation $\prec_k$, maintain a class k and the packet-counter variables $\{R_i^{(k)}\}$ at every node $i = 1, 2, \ldots, |V|$.
3: Each class observes intra-class packet forwarding constraints **(1), (2)** and **(3)** described in sections III and IV.
4: Define the state variables $\{\boldsymbol{Q}^k(t), \boldsymbol{X}^k(t)\}$ and compute the weights $\{\boldsymbol{W}^k(t)\}$, for each class $k = 1, 2, \ldots, K$ exactly as in Eqn. (16), where each class k considers the edges E^k only for Eqns. (15) and (16).
5: An incoming packet to source r at time t joins the class l corresponding to

$$\arg \min_{l \in \mathcal{K}} \sum_{j \in K_r^l(t)} X_j^l(t) \qquad (23)$$

6: The overall weight for an edge e (taken across all the classes) is computed as

$$W_e(t) = \max_{k:e \in E^k} W_e^k(t) \qquad (24)$$

7: Activate the edges corresponding to the max-weight activation, i.e.,

$$\boldsymbol{s}(t) \in \arg \max_{\boldsymbol{s} \in \mathcal{S}} \sum_{e \in E} c_e s_e W_e(t). \qquad (25)$$

8: For each activated edge $e \in \boldsymbol{s}(t)$, forward packets corresponding to a class achieving the maximum in Eqn. (24).

Since the broadcast-rate λ^K achievable by a collection of K embedded DAGs in a graph $\mathcal{G}$ is always upper-bounded by the actual broadcast capacity λ^* of $\mathcal{G}$, we have the following interesting combinatorial result from Theorem (6)

Corollary 2. *Consider a wired network, represented by the graph $\mathcal{G}(V, E)$. For a given integer $K \geq 1$, consider K classes as in Theorem (6), with $\{E^k\}_{k=1}^{K}$ being their corresponding edge-sets. Then, for any set of non-negative vectors $\{\boldsymbol{\beta}^k\}_{k=1}^{K}$ with $\sum_k \beta_{ij}^k \leq 1, \forall (i,j)$, the following lower-bound for the broadcast capacity λ^* holds:*

$$\lambda^* \geq \sum_{k=1}^{K} \min_{j \neq r} \sum_i c_{ij} \beta_{ij}^k \qquad (28)$$

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

The above corollary may be contrasted with Eqn. (7), which provides an upper bound to the broadcast capacity λ^*.

VII. SIMULATION RESULTS

We present a number of simulation results concerning the delay performance of the optimal broadcast policy π^* in wireless DAG networks with different topologies. For simplicity,

we assume primary interference constraints throughout this section. Delay for a packet is defined as the number of slots required for it to reach *all* nodes in the network, after its arrival to the source r.

Diamond topology

We first consider a 4-node diamond topology as shown Fig. 6. Link capacities are shown along with 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. 5(a) 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.

Mesh topology

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 $\{i, j\}$, $1 \le i < j \le 10$, 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 $\{\mathcal{T}_i, 1 \le i \le 5\}$, 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.

Multiclass Simulation for Arbitrary Topology

We randomly generate an ensemble of 500 wired networks (not necessarily DAGs), each consisting of $N = 10$ nodes and unit capacity links. By solving the LP corresponding to Eqn. (26), we compute the fraction of the total broadcast capacity
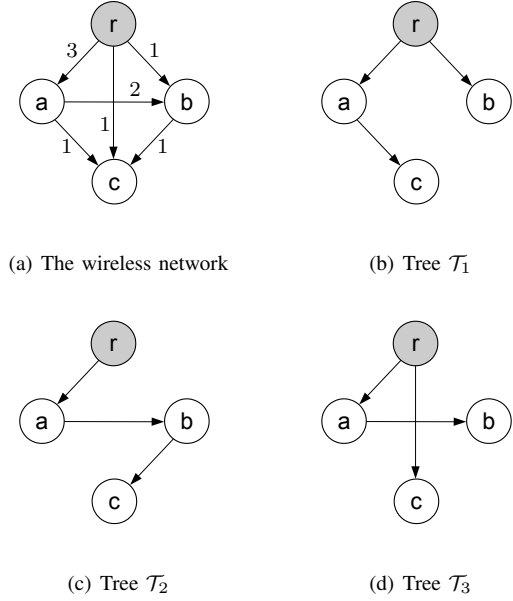

(a) The wireless network   (b) Tree $\mathcal{T}_1$

(c) Tree $\mathcal{T}_2$ (d) Tree $\mathcal{T}_3$

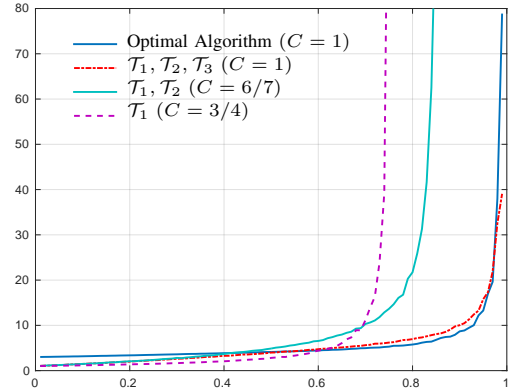Fig. 6: A wireless DAG network and its three embedded spanning trees.

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.

achievable using K randomly chosen classes by the Multiclass Algorithm 2 of section VI. The result is presented in Figure 9. It follows that a sizeable fraction of the optimal capacity may be achieved by using a moderate number of classes. However the number of required classes for achieving a certain fraction of the capacity increases as the broadcast capacity increases. This is because of the fact that increased broadcast capacity would warrant an increased number of DAGs to cover the graph efficiently.

VIII. 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

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^*.
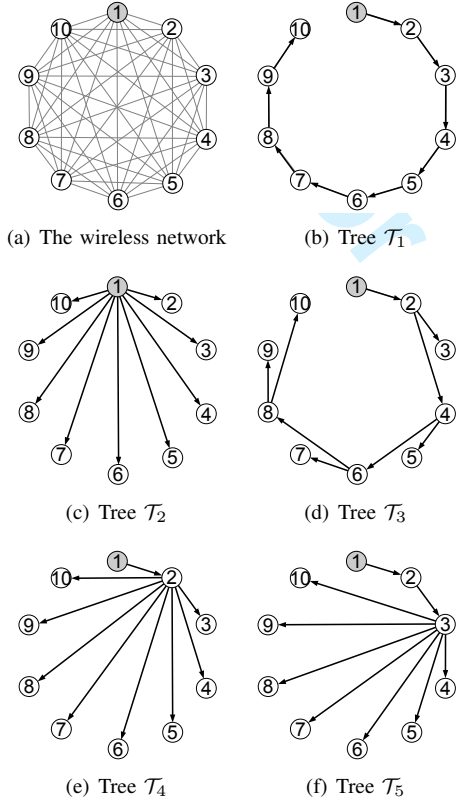

Fig. 8: The 10-node wireless DAG network and a subset of spanning trees.

Fig. 9: 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.

constraint, is promising for application to other systems with packet replicas, such as multicasting and caching systems. Future work involves the study of arbitrary 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] R. Rustin, *Combinatorial Algorithms*. Algorithmics Press, 1973.
[8] 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.
[9] 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.
[10] L. Massoulie, 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.
[11] D. Towsley and A. Twigg, "Rate-optimal decentralized broadcasting: the wireless case," in *ACITA*, 2008.
[12] 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.
[13] 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.
[14] C. Joo, X. Lin, and N. B. Shroff, "Greedy maximal matching: Performance limits for arbitrary network graphs under the node-exclusive interference model," *Automatic Control, IEEE Transactions on*, vol. 54, no. 12, pp. 2734–2744, 2009.
[15] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.
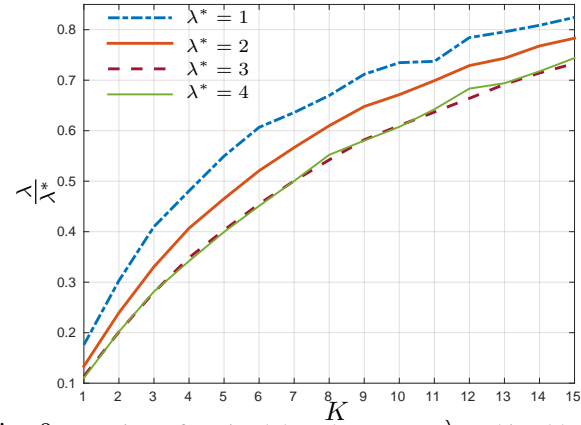
[16] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization.* Athena Scientific Belmont, MA, 1997, vol. 6.

[17] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency.* Springer Science & Business Media, 2003, vol. 24.

[18] J. Matoušek, *Lectures on discrete geometry.* Springer New York, 2002, vol. 108.

[19] D. P. Bertsekas, A. Nedi, A. E. Ozdaglar *et al.*, "Convex analysis and optimization," 2003.

[20] S. Dasgupta, C. H. Papadimitriou, and U. Vazirani, *Algorithms.* McGraw-Hill, Inc., 2006.

APPENDIX

A. Proof of Theorem 1

Fix an $\epsilon > 0$. Consider a policy $\pi \in \Pi$ that achieves a broadcast rate of at least $\lambda^* - \epsilon$ defined in (1); this policy π exists by the definition of the broadcast capacity λ^* in Definition 2. Consider any proper cut U of the network $\mathcal{G}$. By definition, there exists a node $i \notin U$. Let $s^\pi(t) = (s_e^\pi(t), e \in E)$ be the link-activation vector chosen by policy π in slot t. The maximum number of packets that can be transmitted across the cut U in slot t is at most $\sum_{e \in E_U} c_e s_e^\pi(t)$, which is the total capacity of all activated links across U, and the link subset E_U is given in (3). The number of distinct packets received by a node i by time T is upper bounded by the total available capacity across the cut U up to time T, subject to link-activation decisions of policy π. That is, we have

$$R_i^\pi(T) \le \sum_{t=1}^{T} \sum_{e \in E_U} c_e s_e^\pi(t) = \boldsymbol{u} \cdot \sum_{t=1}^{T} \boldsymbol{s}^\pi(t), \quad (29)$$

where we define the vector $\boldsymbol{u} = (u_e, e \in E)$, $u_e = c_e 1_{[e \in E_U]}$, and $\boldsymbol{a} \cdot \boldsymbol{b}$ is the inner product of two vectors.[6] Dividing both sides by T yields

$$\frac{R_i^\pi(T)}{T} \le \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right).$$

It follows that

$$\lambda^* - \epsilon \overset{(a)}{\le} \min_{j \in V} \liminf_{T \to \infty} \frac{R_j^\pi(T)}{T} \le \liminf_{T \to \infty} \frac{R_i^\pi(T)}{T}$$

$$\le \liminf_{T \to \infty} \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right), \quad (30)$$

where (a) follows that π is a broadcast policy of rate at least $\lambda^* - \epsilon$. Since the above holds for any proper-cut $\boldsymbol{u} \in U$, we have

$$\lambda^* - \epsilon \le \min_{\boldsymbol{u} \in U} \liminf_{T \to \infty} \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right) \quad (31)$$

Now consider the following lemma.

Lemma 4. *For any policy $\pi \in \Pi$, there exists a vector $\boldsymbol{\beta}^\pi \in conv(\mathcal{S})$ such that*

$$\min_{\boldsymbol{u} \in U} \liminf_{T \to \infty} \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right) = \min_{\boldsymbol{u} \in U} \boldsymbol{u} \cdot \boldsymbol{\beta}^\pi \quad w.p.1$$

Proof: Consider the sequence $\boldsymbol{\zeta}_T^\pi = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t)$ indexed by $T \ge 1$. Since $\boldsymbol{s}^\pi(t) \in \mathcal{S}$ for all $t \ge 1$, we have

[6]Note that (29) remains valid if network coding operations are allowed.

$\boldsymbol{\zeta}_T^\pi \in \text{conv}(\mathcal{S})$ for all $T \ge 1$. Since $|U|$ is finite, by the definition of $\liminf$, there exists a subsequence $\{\boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_k}^\pi\}_{k \ge 1}$ of the sequence $\{\boldsymbol{u} \cdot \boldsymbol{\zeta}_T^\pi\}_{T \ge 1}$ such that

$$\min_{\boldsymbol{u} \in U} \lim_{k \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_k}^\pi = \min_{\boldsymbol{u} \in U} \liminf_{T \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_T^\pi. \quad (32)$$

Since the set $\text{conv}(\mathcal{S}) \subset \mathbb{R}^{|E|}$ is closed and bounded, by the Heine-Borel theorem, it is compact. Hence any sequence in $\text{conv}(\mathcal{S})$ has a converging sub-sequence. Thus, there exists a sub-sub-sequence $\{\boldsymbol{\zeta}_{T_{k_i}}^\pi\}_{i \ge 1}$ and $\boldsymbol{\beta}^\pi \in \text{conv}(\mathcal{S})$ such that

$$\boldsymbol{\zeta}_{T_{k_i}}^\pi \to \boldsymbol{\beta}^\pi, \quad \text{as } i \to \infty.$$

It follows that

$$\min_{\boldsymbol{u} \in U} \boldsymbol{u} \cdot \boldsymbol{\beta}^\pi \overset{(a)}{=} \min_{\boldsymbol{u} \in U} \lim_{i \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_{k_i}}^\pi$$

$$\overset{(b)}{=} \min_{\boldsymbol{u} \in U} \lim_{k \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_k}^\pi$$

$$\overset{(c)}{=} \min_{\boldsymbol{u} \in U} \liminf_{T \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_T^\pi$$

$$= \min_{\boldsymbol{u} \in U} \liminf_{T \to \infty} \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right),$$

where (a) uses the fact that if $\boldsymbol{x}_n \to \boldsymbol{x}$ then $\boldsymbol{c} \cdot \boldsymbol{x}_n \to \boldsymbol{c} \cdot \boldsymbol{x}$ for $\boldsymbol{c}$, $\boldsymbol{x}_n$, and $\boldsymbol{x} \in \mathbb{R}^l$, $l \ge 1$; (b) follows that if the limit of a sequence $\{z_n\}$ exists then all subsequences $\{z_{n_k}\}$ converge and $\lim_k z_{n_k} = \lim_n z_n$; (c) follows from Equation (32). This completes the proof of the lemma. ∎

Combining Lemma 4 with Eqn. (31), we have that there exists a vector $\boldsymbol{\beta}^\pi \in \text{conv}(\mathcal{S})$ such that

$$\lambda^* - \epsilon \le \min_{\boldsymbol{u} \in U} \boldsymbol{u} \cdot \boldsymbol{\beta}^\pi. \quad (33)$$

Maximizing the right hand side of Eqn. 33 over all $\boldsymbol{\beta}^\pi \in \text{conv}(\mathcal{S})$, we have

$$\lambda^* - \epsilon \le \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \left(\min_{\boldsymbol{u} \in U} \boldsymbol{u} \cdot \boldsymbol{\beta} \right) \quad (34)$$

Since the above inequality holds for any $\epsilon > 0$, by taking $\epsilon \searrow 0$ and expanding the dot product, we have

$$\lambda^* \le \max_{\boldsymbol{\beta} \in \text{conv}(\mathcal{S})} \left(\min_{\boldsymbol{u} \in U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e \right). \quad (35)$$

B. Proof of Lemma 1

Consider the wired network in Fig. 10(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 $\lambda^* \le 2$. In fact, the network has two edge-disjoint spanning trees as shown in Figures 10(b) and 10(c). We can achieve the broadcast capacity $\lambda^* = 2$ by routing odd-numbered and even-numbered packets along the trees $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively.

Consider a policy $\pi \in \Pi_{\text{in-order}}$ that ensures in-order delivery of packets to all nodes. Let $R_i(t)$ be the number of distinct packets received by node i up to time t. Hence, node i receives the set of packets $\{1, 2, \ldots, R_i(t)\}$ by time t due to in-order packet delivery. Consider the directed cycle $a \to b \to c \to a$ in Fig. 10(a). The necessary condition for all links in the cycle to

(a) A wired network with a directed cycle $a \rightarrow b \rightarrow c \rightarrow a$.

(b) Tree $\mathcal{T}_1$
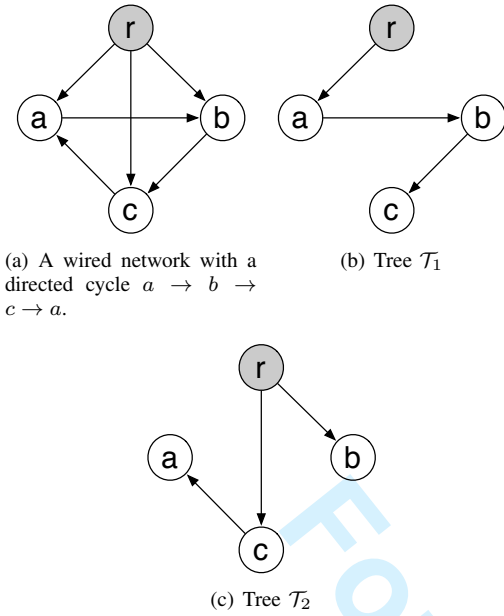
(c) Tree $\mathcal{T}_2$

Fig. 10: A wired network and its two edge-disjoint spanning trees that yield the broadcast capacity $\lambda^* = 2$.

forward (non-duplicate) 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 at 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

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

Taking a $\limsup$ at both sides, we obtain

$$\sum_{e \in \{(a,b),(b,c),(c,a)\}} \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} x_e(t)$$

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

The above inequality implies that

$$\max_{e \in \{(a,b),(b,c),(c,a)\}} \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} x_e(t) \geq \frac{1}{3}. \qquad (36)$$

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 (36), i.e.,

$$\limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} x_{(a,b)}(t) \geq \frac{1}{3}. \qquad (37)$$

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

$$R_b(T) \leq \sum_{t=1}^{T} \left(1 - x_{(r,b)}(t) + 1 - x_{(a,b)}(t) \right)$$

$$\leq \sum_{t=1}^{T} \left(2 - x_{(a,b)}(t) \right).$$

It follows that

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

where the last inequality uses (37). Thus, we have

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

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

$$\lambda^*_{\text{in-order}} = \sup_{\pi \in \Pi_{\text{in-order}}} \min_{i \in V} \liminf_{T \to \infty} \frac{R_i(T)}{T} \leq \frac{5}{3} < 2 = \lambda^*.$$

i.e., the network broadcast capacity is strictly reduced by in-order packet delivery in the cyclic network in Fig. 10(a).

C. Proof of Theorem 2

We present the proof in three steps. First, using the dynamics of $X_j(t)$ in (14), we derive an expression of the one-slot Lyapunov drift using quadratic Lyapunov functions. Second, we design a useful stationary randomized policy that yields ϵ-optimal broadcast throughput; this policy is useful to show that the system $\boldsymbol{X}(t)$, under the optimal broadcast policy π^*, is strongly stable for all arrival rates $\lambda < \lambda^*$. Third, based on the above analysis, we show that the policy π^* is a throughput-optimal broadcast policy for any underlying network graph which is a DAG.

Lemma 5. *If we have*

$$Q(t + 1) \leq (Q(t) - \mu(t))^+ + A(t) \qquad (38)$$

where all the variables are non-negative and $(x)^+ = \max\{x, 0\}$, *then*

$$Q^2(t+1) - Q^2(t) \leq \mu^2(t) + A^2(t) + 2Q(t)(A(t) - \mu(t)).$$

Proof: Squaring both sides of (38) yields

$$Q^2(t+1)$$
$$\leq \left((Q(t) - \mu(t))^+ \right)^2 + A^2(t) + 2A(t)(Q(t) - \mu(t))^+$$
$$\leq (Q(t) - \mu(t))^2 + A^2(t) + 2A(t)Q(t),$$

where we use the fact that $x^2 \geq (x^+)^2$, $Q(t) \geq 0$, and $\mu(t) \geq 0$. Rearranging the above inequality finishes the proof. ∎

Applying Lemma 5 to the dynamics (14) of $X_j(t)$ yields, for each node $j \neq r$,

$$X_j^2(t+1) - X_j^2(t)$$

$$\leq B(t) + 2X_j(t)\Big(\sum_{m\in V}\mu_{mi_t^*}(t) - \sum_{k\in V}\mu_{kj}(t)\Big), \quad (39)$$

where $B(t) \leq \mu_{\max}^2 + \max\{a^2(t), \mu_{\max}^2\} \leq (a^2(t) + 2\mu_{\max}^2)$, $a(t)$ is the number of exogenous packet arrivals in a slot, and $\mu_{\max} \triangleq \max_{e\in E} c_e$ is the maximum capacity of the links. We assume the arrival process $a(t)$ has bounded second moments; thus, there exists a finite constant $B > 0$ such that $\mathbb{E}[B(t)] \leq \mathbb{E}(a^2(t)) + 2\mu_{\max}^2 < B$.

We define the quadratic Lyapunov function $L(\boldsymbol{X}(t)) = \sum_{j\neq r} X_j^2(t)$. From (39), the one-slot Lyapunov drift $\Delta(\boldsymbol{X}(t))$ satisfies

$$\Delta(\boldsymbol{X}(t)) \triangleq \mathbb{E}[L(\boldsymbol{X}(t+1) - L(\boldsymbol{X}(t)) \mid \boldsymbol{X}(t)]$$

$$= \mathbb{E}\Big[\sum_{j\neq r}\big(X_j^2(t+1) - X_j^2(t)\big) \mid \boldsymbol{X}(t)\Big]$$

$$\leq B|V| + 2\sum_{j\neq r}X_j(t)\mathbb{E}\Big[\sum_{m\in V}\mu_{mi_t^*}(t) - \sum_{k\in V}\mu_{kj}(t) \mid \boldsymbol{X}(t)\Big]$$

$$= B|V| - 2\sum_{(i,j)\in E}\mathbb{E}[\mu_{ij}(t) \mid \boldsymbol{X}(t)]\Big(X_j(t) - \sum_{k\in K_j(t)}X_k(t)\Big)$$

$$= B|V| - 2\sum_{(i,j)\in E}\mathbb{E}[\mu_{ij}(t) \mid \boldsymbol{X}(t)]\,W_{ij}(t), \quad (40)$$

where $K_j(t)$ and $W_{ij}(t)$ are defined in (15) and (16), respectively. We emphasize that the evaluation of the inequality (40) depends on a control policy $\pi \in \Pi^*$, we rewrite (40) as

$$\Delta^\pi(\boldsymbol{X}(t)) \leq B|V| - 2\sum_{(i,j)\in E}\mathbb{E}[\mu_{ij}^\pi(t) \mid \boldsymbol{X}(t)]\,W_{ij}(t). \quad (41)$$

Our optimal broadcast policy π^* is chosen to minimize the drift on the right-hand side of (41) among all policies in Π^*.

Next, we construct a randomized scheduling policy $\pi^{\text{RAND}} \in \Pi^*$. Let $\boldsymbol{\beta}^* \in \text{conv}(\mathcal{S})$ be the vector that attains the outer bound on the broadcast capacity λ^* in Theorem 1, i.e.,

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

From Caratheodory's theorem [18], there exist at most $(|E|+1)$ link-activation vectors $\boldsymbol{s}_l \in \mathcal{S}$ and the associated non-negative scalars $\{p_l\}$ with $\sum_{l=1}^{|E|+1}p_l = 1$, such that

$$\boldsymbol{\beta}^* = \sum_{l=1}^{|E|+1}p_l\boldsymbol{s}_l. \quad (42)$$

Hence, from Theorem 1 we have,

$$\lambda^* \leq \min_{U:\text{ a proper cut}}\sum_{e\in E_U}c_e\beta_e^*. \quad (43)$$

Suppose that the exogenous packet arrival rate λ is strictly less than the broadcast capacity λ^*. There exists an $\epsilon > 0$ such that $\lambda + \epsilon \leq \lambda^*$. From (43), we have

$$\lambda + \epsilon \leq \min_{U:\text{ a proper cut}}\sum_{e\in E_U}c_e\beta_e^*. \quad (44)$$

For any network node $v \neq r$, consider the proper cuts $U_v = V \setminus \{v\}$. We have, from (44), that

$$\lambda + \epsilon \leq \sum_{e\in E_{U_v}}c_e\beta_e^*, \quad \forall v \neq r. \quad (45)$$

Since the underlying network topology $\mathcal{G} = (V, E)$ is a DAG, there exists a topological ordering of the network nodes so that: (i) the nodes can be labelled serially as $\{v_1, \ldots, v_{|V|}\}$, where $v_1 = r$ is the source node with no in-neighbours and $v_{|V|}$ has no outgoing neighbours and (ii) all edges in E are directed from $v_i \to v_j$, $i < j$ [20]; From (45), we define $q_l \in [0, 1]$ for each node v_l such that

$$q_l\sum_{e\in E_{U_{v_l}}}c_e\beta_e^* = \lambda + \epsilon\frac{l}{|V|}, \quad l = 2, \ldots, |V|. \quad (46)$$

Consider the randomized broadcast policy $\pi^{\text{RAND}} \in \Pi^*$ working as follows: (i) it selects the feasible link-activation vector $\boldsymbol{s}(t) = \boldsymbol{s}_l$ with probability p_l in (42), $l = 1, 2, \ldots, |E|+1$, in every slot t; (ii) for each selected link $e = (\cdot, v_l)$ of node v_l such that $s_e(t) = 1$, the link e is activated independently with probability q_l; (iii) activated links are used to forward packets, subject to the constraints that define the policy class Π^* (i.e., in-order packet delivery and that a network node is only allowed to receive packets that have been received by all of its in-neighbors). Note that this randomized policy is independent of the state $\boldsymbol{X}(t)$. Since each network node j is relabeled as v_l for some l, from (46) we have, for each node $j \neq r$, the total expected incoming transmission rate satisfies

$$\sum_{i:(i,j)\in E}\mathbb{E}[\mu_{ij}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)] = \sum_{i:(i,j)\in E}\mathbb{E}[\mu_{ij}^{\pi^{\text{RAND}}}(t)]$$

$$= q_l\sum_{e\in E_{U_{v_l}}}c_e\beta_e^*$$

$$= \lambda + \epsilon\frac{l}{|V|}. \quad (47)$$

Equation (47) shows that the randomized policy π^{RAND} provides each network node $j \neq r$ with the total expected incoming capacity strictly larger than the packet arrival rate λ via proper random link activations. According to the abuse of notation in (14), at the source node r we have

$$\sum_{i:(i,r)\in E}\mathbb{E}[\mu_{ir}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)] = \mathbb{E}\Big[\sum_{i:(i,r)\in E}\mu_{ir}^{\pi^{\text{RAND}}}(t)\Big] = \lambda. \quad (48)$$

From (47) and (48), if node i appears before node j in the aforementioned topological ordering, i.e., $i = v_{l_i} < v_{l_j} = j$ for some $l_i < l_j$, then

$$\sum_{k:(k,i)\in E}\mathbb{E}[\mu_{ki}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)] - \sum_{k:(k,j)\in E}\mathbb{E}[\mu_{kj}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)]$$

$$\leq -\frac{\epsilon}{|V|}. \quad (49)$$

The drift inequality (40) holds for any policy $\pi \in \Pi^*$. Our broadcast policy π^* observes the system states $\boldsymbol{X}(t)$ and seek to minimize the drift at every slot. Comparing the actions taken by the policy π^* with those by the randomized policy π^{RAND} in slot t in (40), we have

$$\Delta^{\pi^*}(\boldsymbol{X}(t)) \leq B|V| - 2\sum_{(i,j)\in E}\mathbb{E}\big[\mu_{ij}^{\pi^*}(t) \mid \boldsymbol{X}(t)\big]W_{ij}(t)$$

$$\leq B|V| - 2\sum_{(i,j)\in E}\mathbb{E}\big[\mu_{ij}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)\big]W_{ij}(t)$$

$$
\begin{aligned}
&= B|V| + 2\sum_{j \neq r} X_j(t)\left(\sum_{m \in V}\mathbb{E}\big[\mu_{mi_t^*}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)\big]\right.\\
&\qquad\left. - \sum_{k \in V}\mathbb{E}\big[\mu_{kj}^{\pi^{\text{RAND}}}(t)|\boldsymbol{X}(t)\big]\right)\\
&\leq B|V| - \frac{2\epsilon}{|V|}\sum_{j \neq r} X_j(t). \tag{50}
\end{aligned}
$$

Note that $i_t^* = \arg\min_{i \in \text{In}(j)} Q_{ij}(t)$ for a given node j. Since node i_t^* is an in-neighbour of node j, i_t^* must lie before j in any topological ordering of the DAG. Hence, the last inequality of of (50) follows directly from (49). Taking expectation in (50) with respect to $\boldsymbol{X}(t)$, we have

$$
\mathbb{E}\big[L(\boldsymbol{X}(t+1))\big] - \mathbb{E}\big[L(\boldsymbol{X}(t))\big] \leq B|V| - \frac{2\epsilon}{|V|}\mathbb{E}||\boldsymbol{X}(t)||_1,
$$

where $||\cdot||_1$ is the ℓ_1-norm of a vector. Summing the above over $t = 0, 1, 2, \ldots T-1$ yields

$$
\mathbb{E}\big[L(\boldsymbol{X}(T))\big] - \mathbb{E}\big[L(\boldsymbol{X}(0))\big] \leq B|V|T - \frac{2\epsilon}{|V|}\sum_{t=0}^{T-1}\mathbb{E}||\boldsymbol{X}(t)||_1.
$$

Dividing the above by $2T\epsilon/|V|$ and using $L(\boldsymbol{X}(t)) \geq 0$, we have

$$
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}||\boldsymbol{X}(t)||_1 \leq \frac{B|V|^2}{2\epsilon} + \frac{|V|\,\mathbb{E}[L(\boldsymbol{X}(0))]}{2T\epsilon}
$$

Taking a $\limsup$ of both sides yields

$$
\limsup_{T \to \infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j \neq r}\mathbb{E}[X_j(t)] \leq \frac{B|V|^2}{2\epsilon} \tag{51}
$$

which implies that all virtual-queues $X_j(t)$ are strongly stable.

Next, we show that the strong stability of the virtual queues $X_j(t)$ implies that the policy π^* achieves the broadcast capacity λ^*, i.e., for all arrival rates $\lambda < \lambda^*$, we have

$$
\lim_{T \to \infty}\frac{R_j(T)}{T} = \lambda, \ \forall j.
$$

Equation (14) shows that the virtual queues $X_j(t)$ have bounded departures (due to the finite link capacities). Thus, strong stability of $X_j(t)$ implies that all virtual queues $X_j(t)$ are rate stable [6, Theorem 2.8], i.e., $\lim_{T \to \infty} X_j(T)/T = 0, a.s.$ for all j. It follows that,

$$
\lim_{T \to \infty}\frac{\sum_{j \neq r} X_j(T)}{T} = 0, \quad \text{w.p. } 1 \tag{52}
$$

Now consider any node $j \neq r$ in the network. We can construct a simple path $\sigma(r = u_n \to u_{n-1}\ldots \to u_1 = j)$ from the source node r to the node j by running the following algorithm on the underlying graph $\mathcal{G}(V, E)$.

This algorithm chooses the parent of a node u in the path σ as the one that has the least relative packet deficit as compared to u. Since the underlying graph $\mathcal{G}(V, E)$ is a connected DAG (i.e., there is a path from the source to every other node in the network), the above path construction algorithm always terminates with a path $\sigma(r \to j)$. The number of distinct packets received by node j up to time T can be written as a telescoping sum of relative packet deficits along the path σ,

---

Algorithm 3 $r \to j$ Path Construction Algorithm

Require: Graph $\mathcal{G}(V, E)$, node $j \in V$
1: $i \leftarrow 1$
2: $u_i \leftarrow j$
3: **while** $u_i \neq r$ **do**
4: $\quad u_{i+1} \leftarrow \arg\min_{k \in \text{In}(u_i)} Q_{ku_i}(t)$; ties are broken arbitrarily.
5: $\quad i \leftarrow i + 1$
6: **end while**

---

i.e.,

$$
\begin{aligned}
R_j(T) &= R_{u_1}(T)\\
&= \sum_{i=1}^{n-1}\big(R_{u_i}(T) - R_{u_{i+1}}(T)\big) + R_{u_n}(T)\\
&= -\sum_{i=1}^{n-1} X_{u_i}(T) + R_r(T)\\
&= -\sum_{i=1}^{n-1} X_{u_i}(T) + \sum_{t=0}^{T-1} A(t), \tag{53}
\end{aligned}
$$

where the third equality follows the observation that (see (10))

$$
X_{u_i}(T) = Q_{u_{i+1}u_i}(T) = R_{u_{i+1}}(T) - R_{u_i}(T).
$$

Using $\sum_{i=1}^{n-1} X_{u_i}(t) \leq \sum_{j \neq r} X_j(t)$, (53) and that $X_j(t)$ are non-negative, we have, for each node j,

$$
\frac{1}{T}\sum_{t=0}^{T-1} A(t) - \frac{1}{T}\sum_{j \neq r} X_j(T) \leq \frac{1}{T}R_j(T) \leq \frac{1}{T}\sum_{t=0}^{T-1} A(t).
$$

Taking a limiting time average and the strong law of large numbers for the arrival process, we have

$$
\lim_{T \to \infty}\frac{R_j(T)}{T} = \lambda, \ \forall j. \quad \text{w.p. } 1
$$

This concludes the proof.

D. Proof of Lemma 3

We regard the DAG G as a wired network in which all links can be activated simultaneously. Theorem 2 and (8) show that the broadcast capacity of the wired network G is

$$
\begin{aligned}
\lambda^* = \lambda_{\text{DAG}} &= \min_{U:\text{ a proper cut}}\sum_{e \in E_U} c_e = \min_{\{U_v, v \neq r\}}\sum_{e \in E_{U_v}} c_e\\
&= \min_{v \in V \setminus \{r\}} d_{\text{in}}(v), \tag{54}
\end{aligned}
$$

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 (8) is the all-one vector $\boldsymbol{\beta} = \mathbf{1}$ and that all links have unity capacity. Edmond's Theorem [7] states that the maximum number of disjoint spanning trees in the directed graph G is

$$
k^* = \min_{U:\text{ a proper cut}}\sum_{e \in E_U} c_e. \tag{55}
$$

Combining (54) and (55) completes the proof.