Throughput-Optimal Multihop Broadcast on Directed Acyclic Wireless Networks

Abhishek Sinha, Georgios Paschos, Chih-Ping Li, and Eytan Modiano, *Fellow, IEEE*

***Abstract*— We study the problem of efficiently disseminating packets in multi-hop wireless networks. At each time slot, the network controller activates a set of non-interfering links and forward selected copies of packets on each activated link. The maximum rate of commonly 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. In this paper, 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 information only, and does not require the use of spanning trees. The principal methodological challenge inherent in this problem is the absence of work-conservation principle due to the duplication of packets, which renders usual queuing modeling inapplicable. We overcome this difficulty by studying relative packet deficits and imposing in-order delivery constraints to every node in the network. We show that in-order delivery is throughput-optimal in DAGs and can be exploited to simplify the design and analysis of optimal algorithms. Our capacity characterization also leads to a polynomial time algorithm for computing the broadcast capacity of any wireless DAG under the primary interference constraints. In addition, 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 a superior delay performance as compared with the traditional tree-based approaches.**

***Index Terms*— Broadcasting, network control.**

Manuscript received July 7, 2015; revised February 9, 2016; accepted June 5, 2016; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor D. Manjunath. Date of publication July 9, 2016; date of current version February 14, 2017. This work was supported by the NSF under Grants CNS-1217048 and AST-1547331, the ONR under Grant N00014-12-1-0064, and the ARO MURI under Grant W911NF-08-1-0238. The work of G. Paschos was supported by the WiNC Project through the National and Community Funds (European Social Fund). Part of the paper appeared in the proceedings of IEEE INFOCOM 2015.

A. Sinha and E. Modiano are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: sinhaa@mit.edu; modiano@mit.edu).

G. Paschos was with the Informatics and Telematics Institute, Centre for Research and Technology Hellas, Thessaloniki 57001, Greece, and also with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with the Mathematical and Algorithmic Sciences Laboratory, France Research Center, Huawei Technologies Company, Ltd., 92100 Boulogne–Billancourt, France (e-mail: georgios.paschos@huawei.com).

C.-P. Li was with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is now with Qualcomm Research, San Diego, CA 92121 USA (e-mail: cpli@qti.qualcomm.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. It consists of a 137-kB PDF.

Digital Object Identifier 10.1109/TNET.2016.2582907

I. Introduction and Related Work

BROADCASTING refers to the fundamental network functionality of delivering data from a source node to all other nodes in a network. For efficient broadcasting, one needs to use appropriate packet replication and forwarding to eliminate redundant transmissions. This is especially important in power-constrained wireless networks which suffer from interference and packet 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 several 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, more so when this is to be done repeatedly as and when the 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 maximum achievable common reception rate of distinct packets over all scheduling policies is known as the broadcast capacity of the network. To the best of our knowledge, no capacity-achieving broadcast policy for wireless networks is known that does not use spanning trees. The main contribution of this paper is to design a decentralized and provably optimal wireless broadcast algorithms that does not use spanning trees when the underlying network topology is restricted to a DAG. Many wireless networks fall in this category [5], [6].

To design the algorithm, 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 tractable set of policies without any loss of throughput-optimality. First, we consider the subclass of policies $\Pi^{\text{in-order}} \subset \Pi$ which delivers packets to all nodes, in the same order they arrive at the source, i.e., *in-order*.

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 apparent that the policies in the more structured class Π^* are easier to describe and analyze, but might not be throughput-optimal. We prove the surprising result that when the underlying network topology is a directed acyclic graph (DAG), there is a broadcast policy $\pi^* \in \Pi^*$ that achieves the broadcast capacity of the network. In contrast, we also find a (non-DAG) network containing a directed cycle in which *no* control policy in the space $\Pi^{\text{in-order}}$ can provably achieve the broadcast capacity.

To design the optimal broadcast policy π^*, we first 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 structures are difficult to define due to packet duplications and consequent loss of work-conservation. We subsequently show that, the problem of achieving the broadcast capacity in a DAG reduces to the problem of finding a scheduling policy *stabilizing* the relative packet deficits, which can be solved by utilizing Lyapunov drift analysis techniques [7], [8].

In this paper, our main contributions are as follows:

- We define the broadcast capacity of a wireless network and show that it can be 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 $\mathcal{G}$, which is also equal to the maximum number of edge-disjoint directed spanning trees rooted at the source.
- 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 wireless networks with time-varying topology. This algorithm also yields a constructive proof of a version of Edmonds' disjoint tree-packing theorem [9] 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 the primary interference constraints.
- We propose a randomized multiclass extension of our broadcast algorithm, which can be effectively used to do broadcast on wireless networks with arbitrary topology.
- We demonstrate the superior delay performance of our algorithm, as compared to the centralized tree-based algorithms [4], via numerical simulations. We also explore the efficiency/complexity trade-off of the proposed multiclass extension through extensive numerical simulations.

Related Works: In the literature, a simple method for wireless broadcast is to use packet flooding [10]. The flooding approach, however, leads to redundant transmissions and collisions, known as *broadcast storm* [11]. In the wired domain, it has been shown that forwarding *useful* packets at random is optimal for broadcast [12]; this approach, however, does not extend to the wireless setting due to interference and the need for scheduling appropriate activation sets [13]. Broadcasting on wire line networks can also be carried out using network coding [14], [15]. However, efficient link activation under network coding remains an open problem. There are also a number of papers on minimizing the total *latency* for broadcasting a finite number of packets in a network [16]–[18]. However these works do not deal with achieving the capacity of the network, which is the focus of this paper.

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 outer bound on the capacity from a cut-set consideration. In Section IV, we propose a dynamic broadcast algorithm 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 the 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 represented by the tuple $\big(\mathcal{G}(V,E), \boldsymbol{c}, \mathcal{S}\big)$, where V is the set of nodes, E is the set of directed point-to-point 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-activations. An element $\boldsymbol{s} = (s_e, e \in E)$ of the activation set $\mathcal{S}$ is an $|E|$-dimensional binary vector, such that all links $e \in E$ with $s_e = 1$ can be activated simultaneously at a slot. 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* [19]), the set $\mathcal{S}$ consists of $|E|$-dimensional binary vectors corresponding to different *matchings* of the underlying graph $\mathcal{G}$ [20], see Fig. 1. For the case of a wire line 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 $\mathcal{S}$, thus capturing arbitrary wireless interference models.

We note that, some wireless networks might have *point-to-multi-point links*, where a transmission can be heard by all out-neighbours. In this case, network-capacity expression and analysis of optimal algorithms would be different. In this paper, for simplicity, we do not consider such situations.

Let $\mathtt{r} \in V$ be the source node at which stochastic broadcast traffic is generated (or arrives externally). The number of packets generated at the node $\mathtt{r}$ at slot t is denoted by the random variable $A(t) \in \mathbb{Z}_+$, which is i.i.d. over slots with mean λ and bounded second moment. These packets are to be disseminated efficiently to all other nodes in the network.

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

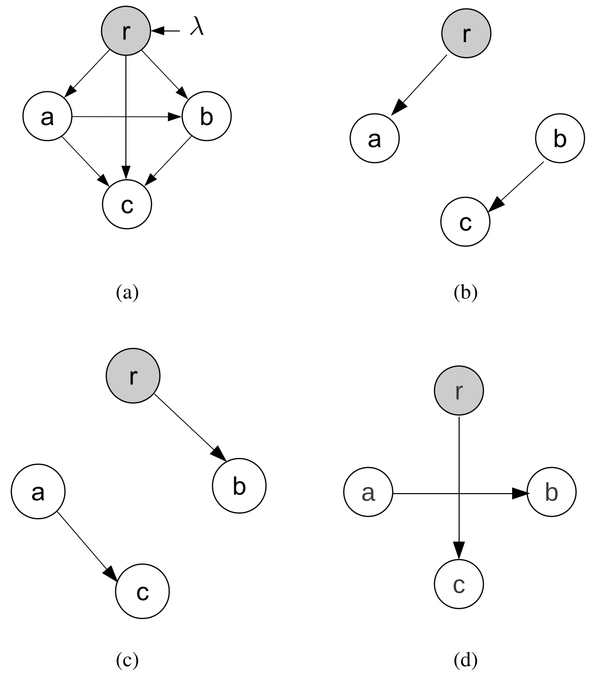

Fig. 1. A wireless network and its three feasible link activations under the primary interference constraint. (a) A wireless network. (b) Activation vector $\boldsymbol{s}_1$. (c) Activation vector $\boldsymbol{s}_2$. (d) Activation vector $\boldsymbol{s}_3$.

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 $e = (i, j)$ (with $\boldsymbol{s}_e(t) = 1$), subject to the link capacity constraint. The policy class Π includes policies that may 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 the action of 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 (Broadcast Policy): A policy $\pi \in \Pi$ is called a "broadcast policy of rate λ" if all nodes receive distinct packets at rate λ, i.e.,

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

where λ is the packet arrival rate at the source node $\mathtt{r}$.

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 scheduling policies in Π. We first give an intuitive explanation of the bound, assuming that the limits involved exist. In the proof of Theorem 1 we rigorously establish the result without 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; i.e., define the vector

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

where $\boldsymbol{s}^\pi(t)$ is the chosen link-activation vector by policy π in slot t. The average packet 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 associated edge-capacitated graph, $\widehat{\mathcal{G}}^\pi = (V, E, (\widehat{c}_e))$ where each 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 average edge-activation vectors $\boldsymbol{\beta}^\pi$.

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 $\mathtt{r}$. Define the edge-cut E_U associated U as

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

Since $U \subset V$, there exists a node $n \in V \setminus U$. Consider the throughput (rate of packet reception) 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. This cut-set bound is valid even when we consider the general flow of *information* in the network (see [21, Th. 15.10.1]). Hence the cut-set bound holds even when we allow network coding operations. By definition of achievable broadcast rate λ^π, 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

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

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

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

where the last inequality holds because the vector $\boldsymbol{\beta}^\pi$ lies in the convex hull of the activation set $\mathcal{S}$; Refer to Eqn. (2). 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:

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

Proof: See Appendix A. ■

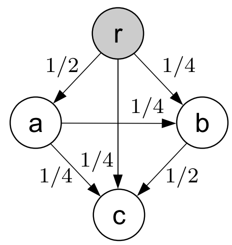

Fig. 2. The edge-capacitated graph $\widehat{\mathcal{G}}^\pi$ for the wireless network with unit link capacities in Fig. 1 and under the time-average vector $\boldsymbol{\beta}^\pi = (1/2, 1/4, 1/4)$. The link weights are the capacities $c_e\beta_e^\pi$. The minimum proper cut in this graph has value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

B. Constrained Policy-Space: In-Order Packet Delivery

Studying the performance of any arbitrary broadcast policy $\pi \in \Pi$ is analytically formidable because packets are replicated across the network and may be received out of order. To avoid unnecessary re-transmissions, each node 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 (as in classical back-pressure algorithm [7]), the system state is properly described here by the identity of the subset of packets received at each of the nodes.

To simplify the state, we focus on the subset of policies $\Pi^{\text{in-order}} \subset \Pi$ that enforce the following constraint. It will be shown subsequently that, this restriction can be made without loss of throughput-optimality in a DAG.

Index the packets serially $\{1, 2, \ldots\}$ according to their order of arrival at the source.

Constraint 1 (In-Order Packet Delivery $\Pi^{\text{in-order}}$): In this policy-space, a node is allowed to receive a packet p at slot t only if all previous packets $\{1, 2, \ldots, p-1\}$ have been received by that node by slot t.

In-order packet delivery is practically useful in live media streaming applications [2] where buffering out-of-order packets incurs increased delay and degrades the playback quality. As shown below, Constraint 1 greatly simplifies the state space representation of the system.

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 show the existence of a throughput-optimal broadcast policy in the space $\Pi^{\text{in-order}}$ when the underlying topology is a DAG. On the other hand, the following complementary result, Lemma 1, says that there exists a non-DAG network in which *any* broadcast policy in the space $\Pi^{\text{in-order}}$ is *not* throughput optimal. This implies that the policy-space $\Pi^{\text{in-order}}$ can not, in general, be utilized 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: See the supplementary material. ■

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 the well-known topological ordering of the nodes [20]. For DAGs, the upper bound (5) on the broadcast capacity λ^* in Theorem 1 will be relaxed further. For each receiver node $v \neq \texttt{r}$, consider the proper cut U_v that separates the node v from the rest of the network. i.e.,

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

Using these collection of cuts $\{U_v, v \neq \texttt{r}\}$, we obtain a relaxed upper bound λ_{DAG} on the broadcast capacity λ^* as:

$$\begin{aligned}\lambda_{\text{DAG}} &\triangleq \max_{\boldsymbol{\beta}\in\text{conv}(\mathcal{S})} \min_{\{U_v, v\neq \texttt{r}\}} \sum_{e\in E_{U_v}} c_e\,\beta_e \qquad (7)\\ &\geq \max_{\boldsymbol{\beta}\in\text{conv}(\mathcal{S})} \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e\,\beta_e \geq \lambda^*,\end{aligned}$$

where the first inequality uses the subset relation $\{U_v, v \neq \texttt{r}\} \subseteq \{U\text{: a proper cut}\}$ 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

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

The capacity 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 topology is a DAG. As a corollary, we also retrieve the result that network-coding operations do not increase the broadcast-capacity in our setting.

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 rest of the network. Note that, there are only $|V| - 1$ of such cuts, in contrast with the total number of cuts in Eqn. (5), 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 wireless DAG network under the primary interference constraints.

IV. DAG Broadcast Algorithm

In this section we design a throughput-optimal broadcast policy for wireless DAGs. We start by imposing an additional constraint on packet-forwarding 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

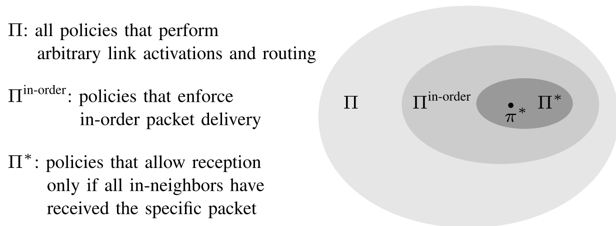

Fig. 3. Containment relationships among different policy classes.

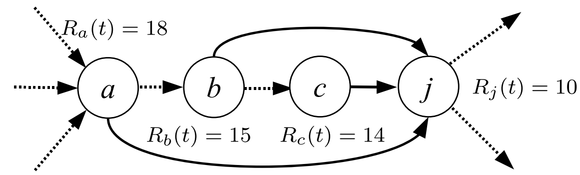

Fig. 4. Under a policy $\pi \in \Pi^*$, the set of packets available for transmission to node j in slot t is $\{11, 12, 13, 14\}$, which are present at all in-neighbors of the node j. The in-neighbor of j having the smallest packet deficit is $i_t^* = c$, and $X_j(t) = \min\{Q_{aj}(t), Q_{bj}(t), Q_{cj}(t)\}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

constitute a simple dynamics. We analyze the dynamics of the minimum relative packet deficits, which behaves like *virtual queues*. We design a dynamic control policy $\pi^* \in \Pi^*$ that stabilizes the virtual queues. 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 earlier in Section III-B that, constrained to the policy-space $\Pi^{\text{in-order}}$, the system-state is completely represented by the vector $\boldsymbol{R}(t)$. However this constrained policy-class alone is not sufficient to obtain a one-step dynamics of the system, which is an essential prerequisite to design a stabilizing control policy. As a result, we restrict our attention to a sub-class of policies in $\Pi^{\text{in-order}}$, defined as follows.

A node i is called an *in-neighbor* of node j if there exists a directed link $(i, j) \in E$ in the underlying graph $\mathcal{G}$. The set of all in-neighbours of a node j is denoted by $\delta_{\text{in}}(j)$. The out-neighbours of a node is defined similarly.

Constraint 2 (Policy-space Π^*): 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 $\Pi^* \subseteq \Pi^{\text{in-order}}$. It will be shown subsequently that this restriction can be done without loss of throughput-optimality. Fig. 3 shows the relationship among different policy classes.[1]

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

Lemma 2: Under any policy $\pi \in \Pi^*$, we have:

(1) $R_j(t) \le \min_{i\in\delta_{\text{in}}(j)} R_i(t)$

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

$$\big\{p \mid R_j(t) + 1 \le p \le \min_{i\in\delta_{\text{in}}(j)} R_i(t)\big\}.$$

The proof of the above lemma follows directly from the definition of the policy-space Π^*.

Define the *packet-deficit* $Q_{ij}(t)$ over the link $(i, j) \in E$ to be $Q_{ij}(t) \stackrel{\text{def}}{=} R_i(t) - R_j(t)$. Under a policy in Π^*, $Q_{ij}(t)$ is always non-negative because, by part (1) of Lemma 2, we have

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

The variable $Q_{ij}(t)$ 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

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

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 under Π^*.

As an example, Fig. 4 shows that the packet deficits at node j, relative to its in-neighbors 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

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

i.e., 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^*.[2] Our optimal broadcast policy will be described in terms of the minimum packet deficits $\{X_j(t)\}_{j\neq \mathtt{r}}$.

B. The Dynamics of the State Variables $\{X_j(t)\}$

We now analyze the dynamics of the state 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 $\mu_{ij}(t) = c_{ij} s_{ij}(t)$, where $\boldsymbol{s}(t)$ is the link-activation vector $\boldsymbol{s}(t)$, chosen for slot t. At node j, the increase in the value of number of packets received, i.e., $R_j(t)$, depends on the identity of the received packets; in particular for efficiency, the node j must receive distinct packets. Next, we clarify which set of packets are allowed to be received by node j at time t.

The number of available packets for reception at node j is $\min\{X_j(t), \sum_{k\in V} \mu_{kj}(t)\}$. This is because: (i) $X_j(t)$ is

[1] 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.

[2] We note that the minimizer i_t^* is a function of the node j and the time slot t and should be properly denoted as $i_t^*(j)$; we slightly abuse the notation by dropping the symbol j from i_t^* throughout to simplify notations.

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)

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

The specific subset of packets that are pulled over each incoming link are disjoint but otherwise arbitrary.[3]

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 per node is activated 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 increase of $Q_{i_t^*j}(t)$ by the total service rate of the activated incoming links at node i_t^*, i.e., $\sum_{m\in V}\ \mu_{mi_t^*}(t)$. 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, 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[4]

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

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

$$\begin{aligned} X_j(t+1) &\overset{(a)}{=} \min_{i\in\delta_{\text{in}}(j)} Q_{ij}(t+1) \overset{(b)}{\le} Q_{i_t^*j}(t+1) \\ &\overset{(c)}{\le} \big(X_j(t) - \sum_{k\in V}\mu_{kj}(t)\big)^+ + \sum_{m\in V}\mu_{mi_t^*}(t), \end{aligned} \tag{14}$$

where the equality (a) follows the definition of $X_j(t)$, inequality (b) follows because node $i_t^* \in \delta_{\text{in}}(j)$ and inequality (c) follows from Eqn. (13). In Eqn. (14), if $i_t^* = \mathtt{r}$, the notation is slightly abused to define $\sum_{m\in V}\ \mu_{mr}(t) = A(t)$ for the source node $\mathtt{r}$, where $A(t)$ is the number of exogenous packet arrival at source at slot t.

C. A Throughput-Optimal Broadcast Policy

Like the Back-Pressure algorithm [7], our broadcast policy is designed to keep the deficit process $\{\boldsymbol{X}(t)\}_{t\ge 0}$ stochastically 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. However, the way the weights are computed in π^* (16), is very much different from the Back-Pressure algorithm. Also the fundamental feature of packet duplications is essentially new here. The policy π^* belongs to the space Π^* and 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. The steps of the algorithm are illustrated in Fig. 5.

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 $K_j(t) \subset \delta_{\text{out}}(j)$ for which node j is their deficit minimizer, i.e.,

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

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 $X_j(t) = \min_{i\in\delta_{\text{in}}(j)} Q_{ij}(t)$ for $j \ne \mathtt{r}$ and assign to link (i,j) the weight

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

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

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

4: Every node $j \ne \mathtt{r}$ 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 the Constraint 3.

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

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

Distributed Implementation: As evident from the description of Algorithm 1, computation of the weight-vectors $\boldsymbol{W}(t)$ and packet forwarding decisions are made locally by individual nodes. The only network-wide operation that the algorithm needs to perform is step 3, where it needs to select the maximum-weighted feasible activation set. The problem of scheduling the Max-weight activation set in a distributed fashion has been studied extensively in the literature [22], [23]. In particular, the work of Bui et al. [23] designs a distributed algorithm for solving the Max-weight scheduling problem with constant overhead in the primary interference setting.

The next theorem demonstrates the optimality of the broadcast policy π^* described above.

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

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

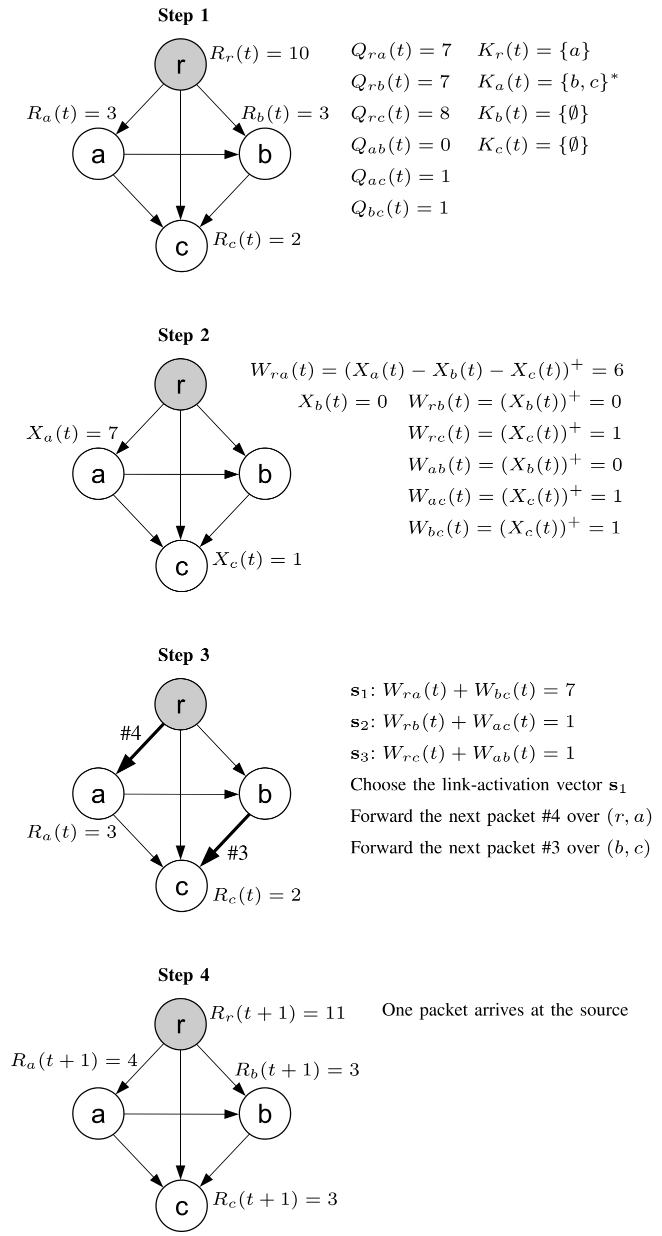

Fig. 5. Running the optimal broadcast policy π^* in slot t in a wireless network with unit-capacity links and under the primary interference constraint. Step 1: computing the deficits $Q_{ij}(t)$ and $K_j(t)$; a tie is broken in choosing node a as the in-neighbor deficit minimizer for node c, hence $c \in K_a(t)$; 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.

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

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

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

Proof: See Appendix B. ∎

D. Number of Disjoint Spanning Trees in a DAG

As a corollary, Theorem 2 yields 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 $\mathtt{r}$, has unit-capacity links, and possibly contains parallel edges. The maximum number k^* of edge-disjoint spanning trees in G is given by

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

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

Proof: See Appendix C. ∎

V. An 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 utilizing the ellipsoid algorithm [24].

Under the primary interference constraint, the set of feasible activations of a graph are its *matchings* [20]. 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

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

We have the following classical result from Edmonds [25].

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

$$\begin{aligned} \beta_e &\geq 0 \quad \forall e \in E \\ \sum_{e\in\delta_{\text{in}}(v)\cup\delta_{\text{out}}(v)} \beta_e &\leq 1 \quad \forall v \in V \\ \sum_{e\in E[U]} \beta_e &\leq \frac{|U|-1}{2}; \quad U \subset V,\ |U| \text{ odd}, \end{aligned} \tag{19}$$

where $E[U]$ is the set of edges (ignoring their directions) with both end points in the subset $U \subset V$.

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

$$\max \lambda \tag{20}$$

Subject to,

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

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

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 an efficient separator for the matching polytope.

Theorem 4 [25]: There exists a strongly poly-time algorithm that, given $\mathcal{G} = (V, E)$ and $\boldsymbol{\beta} : E \to \mathbb{R}^{|E|}$, determines if $\boldsymbol{\beta}$ is feasible for (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 [26], 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: Multiclass Algorithm

In this section we extend the above broadcast policy for DAGs to arbitrary networks, which may possibly contain directed cycles. From the negative result of Lemma 1, we know that any policy ensuring *in-order* packet delivery at every node, cannot achieve the broadcast capacity in arbitrary networks in general. To get around this difficulty, we introduce the notion 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 given permutation $\prec_k$ of the nodes; for an edge $(a, b) \in E$ if the node a appears *before* 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 $\mathcal{G}(V, E)$. Different classes correspond to different permutations of nodes.

An incoming packet at the source node is admitted to some class $k \in \mathcal{K}$, according to some admission-policy. All packets admitted in a given class $k \in \mathcal{K}$ are broadcasted while maintaining the in-order delivery restriction within the class k,

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 $\mathtt{r}$ at time t joins the class k corresponding to

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

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

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

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

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

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

however there is no such inter-class constraint for delivering packets from different classes. Hence the resulting multi-class policy does not belong to the space Π^* but belongs to the general policy-space Π. This new multi-class policy keeps the best from both worlds: (a) its state-space complexity is $\Theta(|\mathcal{K}||V|)$, where for each class we have the same state-representation as in π^* 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 with sufficiently many classes.

Hence the broadcasting problem reduces to construction of multiple classes (equivalently, permutations of the vertices V) in $\mathcal{G}$ such that they cover 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 $\mathtt{r}$ always appears at the first position of the permutation.

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

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

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 k 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 time-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 π^{STAT} is simply $\lambda^K = \sum_{k=1}^K \lambda^k = \sum_k \min_j \sum_i c_{ij}\beta_{ij}^k$. Given these K classes, following the same line of argument as in (20), we can develop a similar LP to compute the broadcast-rate achievable (26) by these K classes by maximizing over all feasible $\{\boldsymbol{\beta}^k\}_1^K$, 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:

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

We then compare the drift of multiclass broadcast algorithm 2 with the stationary randomized 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.

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 as a corollary of Theorem (6)

Corollary 2: Consider a wire line network, represented by the graph $\mathcal{G}(V,E)$. For a given integer $K \geq 1$, consider K arbitrary classes (i.e., permutations of vertices) 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:*

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

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^*. We also note that, the lower-bound in Eqn. (28) is tight when the classes are chosen corresponding to the maximum number of edge-disjoint spanning trees, obtained from Edmonds' Theorem [9].

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 for wireless networks 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 $\mathtt{r}$.

A. Diamond Topology

Consider a 4-node wireless network as shown Fig. 6 (a). Link capacities are indicated alongside the links. The broadcast capacity of the network is upper bounded by the total capacity of incoming links to node $\mathtt{c}$, which is 1. This is because at most one of its unit-capacity incoming links to node $\mathtt{c}$ may be activated at any slot, under the primary interference constraint. To determine the broadcast-capacity of the network, consider three spanning trees $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ rooted at the source node $\mathtt{r}$, as shown in Fig. 6 (b),(c),(d). By finding an optimal time-sharing of all feasible link-activations over a subset of spanning trees using linear programming and using Eqn. (26), we can show that the broadcast-rate achievable using the tree $\mathcal{T}_1$ only is $3/4$, using the trees $\{\mathcal{T}_1, \mathcal{T}_2\}$ only is $6/7$, and using the trees $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ together is 1. Thus, the upper-bound is achieved and the broadcast capacity of the network is $\lambda^* = 1$.

We compare the performance of our throughput-optimal broadcast policy π^* with the tree-based policy π_{tree} proposed

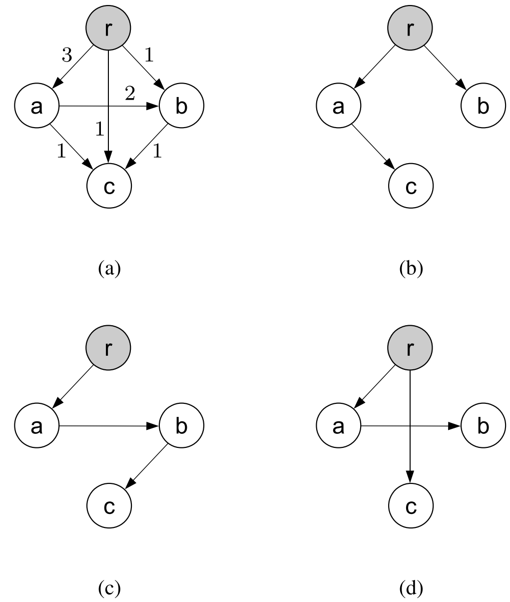

Fig. 6. A wireless DAG network and its three embedded spanning trees. (a) The wireless network. (b) Tree $\mathcal{T}_1$. (c) Tree $\mathcal{T}_2$. (d) Tree $\mathcal{T}_3$.

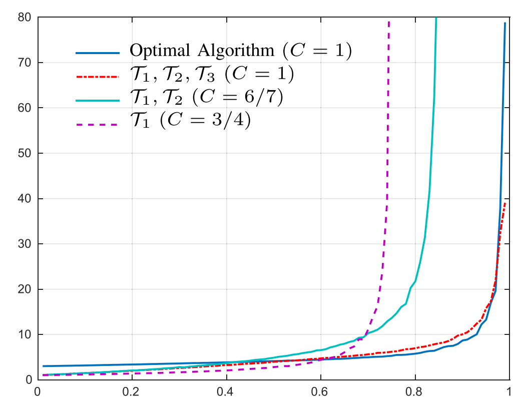

Fig. 7. Average delay performance of the optimal broadcast policy π^* and the tree-based policy π_{tree} that balances traffic over different subsets of spanning trees.

TABLE I

AVERAGE DELAY PERFORMANCE OF THE TREE-BASED POLICY π_{tree} OVER DIFFERENT SUBSETS OF SPANNING TREES AND THE BROADCAST POLICY π^*

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

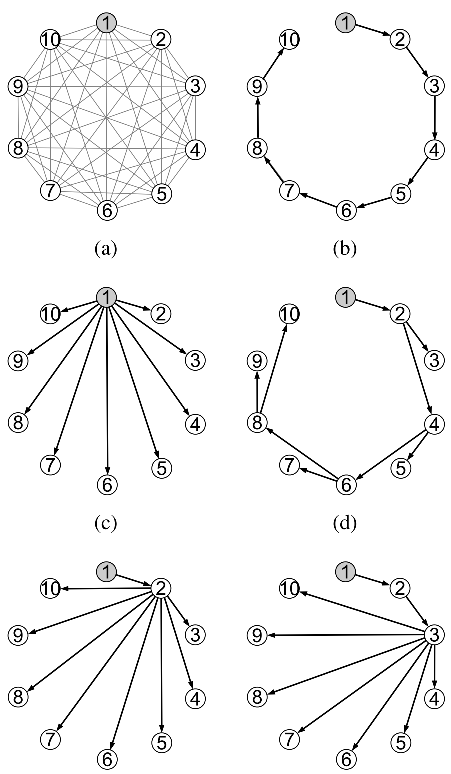

Fig. 8. The 10-node wireless DAG network and a subset of spanning trees. (a) The wireless network. (b) Tree $\mathcal{T}_1$. (c) Tree $\mathcal{T}_2$. (d) Tree $\mathcal{T}_3$. (e) Tree $\mathcal{T}_4$. (f) Tree $\mathcal{T}_5$.

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 generalize their policy π_{tree} for broadcasting packets over spanning trees in the wireless setting. Fig. 5(a) shows 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 and, in general, has better delay performance than the tree-based scheme in the high traffic regime.

B. Mesh Topology

Since the throughput-optimal broadcast policy π^* does not rely on limited number of tree structures, it has the potential to exploit all degrees of freedom in the network. Such freedom leads to better delay performance as compared to the tree-based broadcast policies [4]. To illustrate this effect, consider the 10-node DAG network in Fig. 8 (a). For every pair of node $\{i, j\}$, $1 \leq i < j \leq 10$, the network has a directed link from node i to j with capacity $(10 - i)$. By induction, the number of spanning trees rooted at the source node 1 can be calculated to be $9! \approx 3.6 \times 10^5$. Among them, we choose five arbitrary spanning trees $\{\mathcal{T}_i, 1 \leq i \leq 5\}$, shown in Fig. 8 (b),(c),(d),(e),(f), over which the tree-based algorithm π_{tree} is simulated. Table I demonstrates the superior delay performance of our throughput-optimal broadcast policy π^*, as compared to that of the tree-based algorithm π_{tree}. The table also shows that a tree-based algorithm that does not use enough number of trees might result in degraded broadcast throughput.

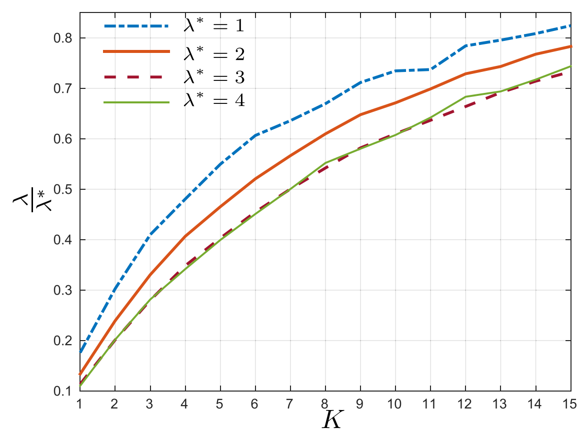

Fig. 9. Fraction of optimal broadcast rate $\frac{\lambda}{\lambda^*}$ achievable by the multiclass broadcast algorithm with randomly chosen K classes for randomly generated wired networks with $N = 10$ nodes.

C. Multiclass Simulation for Arbitrary Topology

To simulate the multiclass broadcast algorithm of section VI, we randomly generate an ensemble of 500 wire line 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 average fraction of the total broadcast capacity achievable using K randomly chosen classes by the Multiclass Algorithm 2 of section VI. The result is plotted in Figure 9. It follows that a sizeable fraction of the optimal capacity may be achieved by using a moderately many classes. However, it also shows that the required number of classes for achieving a certain fraction of the capacity increases as the broadcast capacity of the network increases. This is due to the fact that increased broadcast capacity of a network would warrant an increased number of DAGs to cover it efficiently.

VIII. Conclusion

In this paper we study the problem of broadcasting in a wireless network under general interference constraints. When the underlying network topology is a DAG, we propose a dynamic algorithm that achieves the broadcast capacity of the network. Our novel algorithm, based on packet deficits and the in-order packet delivery constraint, is promising for application to other systems with packet duplications, such as multicasting and caching systems. We also propose a heuristic extension of our DAG broadcast algorithm to networks with arbitrary topology. Future work would involve detailed study of throughput-optimal broadcasting in arbitrary networks, where optimal policies must be sought in the space $\Pi \setminus \Pi^{\text{in-order}}$.

Appendix

A. Proof of Theorem 1

Fix an $\epsilon > 0$. Consider a broadcast policy $\pi \in \Pi$ that achieves a broadcast rate of at least $\lambda^* - \epsilon$, as 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 of a proper-cut, there exists a node $i \notin U$. Let $\boldsymbol{s}^\pi(t) = (s_e^\pi(t), e \in E)$ 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 by any policy 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 the outgoing-edges from the cut U, where the link subset E_U has been defined in Eqn. (3). Thus, the number of distinct packets $R_i^\pi(T)$ received by a node i by time T can be upper-bounded as follows

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

where we define the $|E|$-dimensional cut-vector $\boldsymbol{u} = (u_e, e \in E)$, such that $u_e = c_e 1_{[e \in E_U]}$, and $\boldsymbol{a} \cdot \boldsymbol{b}$ is the inner product of two vectors.[5] Dividing both sides by T yields

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

Hence,

$$\begin{aligned} \lambda^* - \epsilon &\overset{(a)}{\leq} \min_{j \in V} \liminf_{T \to \infty} \frac{R_j^\pi(T)}{T} \leq \liminf_{T \to \infty} \frac{R_i^\pi(T)}{T} \\ &\leq \liminf_{T \to \infty} \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right), \end{aligned} \tag{30}$$

[5] Note that Eqn. (29) remains valid even if the network coding operations are allowed.

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

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

Now consider the following lemma.

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

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

Proof: Consider a sequence of vectors $\boldsymbol{\zeta}_T^\pi \overset{\text{def}}{=} \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t)$, indexed by $T \geq 1$. Since $\boldsymbol{s}^\pi(t) \in \mathcal{S}$ for all $t \geq 1$, we have $\boldsymbol{\zeta}_T^\pi \in \text{conv}(\mathcal{S})$ for all $T \geq 1$. Since $|\mathcal{U}|$ is finite, by the definition of $\liminf$, there exists a sub-sequence $\{\boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_k}^\pi\}_{k \geq 1}$ of the sequence $\{\boldsymbol{u} \cdot \boldsymbol{\zeta}_T^\pi\}_{T \geq 1}$ such that

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

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 $\{\boldsymbol{\zeta}_{T_{k_i}}^\pi\}_{i \geq 1}$ 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

$$\begin{aligned} \min_{\boldsymbol{u} \in \mathcal{U}} \boldsymbol{u} \cdot \boldsymbol{\beta}^\pi &\overset{(a)}{=} \min_{\boldsymbol{u} \in \mathcal{U}} \lim_{i \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_{k_i}}^\pi \\ &\overset{(b)}{=} \min_{\boldsymbol{u} \in \mathcal{U}} \lim_{k \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_k}^\pi \\ &\overset{(c)}{=} \min_{\boldsymbol{u} \in \mathcal{U}} \liminf_{T \to \infty} \boldsymbol{u} \cdot \boldsymbol{\zeta}_T^\pi \\ &\overset{(d)}{=} \min_{\boldsymbol{u} \in \mathcal{U}} \liminf_{T \to \infty} \boldsymbol{u} \cdot \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}^\pi(t) \right), \end{aligned}$$

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 any $\boldsymbol{c}$, $\boldsymbol{x}_n$, and $\boldsymbol{x} \in \mathbb{R}^l$, $l \geq 1$; (b) follows from the fact that if the limit of a sequence $\{z_k \equiv \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_k}^\pi\}$ exists then all sub-sequences $\{z_{k_i} \equiv \boldsymbol{u} \cdot \boldsymbol{\zeta}_{T_{k_i}}^\pi\}$ converge and $\lim_i z_{k_i} = \lim_k z_k$; (c) follows from Equation (32) and (d) follows from the definition of the sequence $\boldsymbol{\zeta}_T^\pi$. 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

$$\lambda^* - \epsilon \leq \min_{\boldsymbol{u} \in \mathcal{U}} \boldsymbol{u} \cdot \boldsymbol{\beta}^\pi. \tag{33}$$

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

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

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

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

B. Proof of Theorem 2

We complete the proof in four steps. First, using the dynamics of $X_j(t)$ in the space Π^* (Eqn. (14)), we derive an expression of one-slot drift of an appropriately defined Quadratic Lyapunov function $L(\boldsymbol{X}(t))$. Second, we design an auxiliary stationary randomized policy π^{RAND} for link-activations that yields optimal broadcast throughput. Third, this randomized policy is used to show that the system $\boldsymbol{X}(t)$ is *strongly stable* for all arrival rates $\lambda < \lambda^*$, under the optimal broadcast policy $\pi^* \in \Pi^*$. Finally, based on the above analysis, we finally show that the policy π^* is a throughput-optimal broadcast policy for any wireless DAG network.

(a) An Upper-bound on the drift of the policy π^*:

Lemma 5: For the dynamics

$$Q(t+1) \leq (Q(t) - \mu(t))^+ + A(t) \tag{36}$$

where all variables are non-negative and $(x)^+ \stackrel{def}{=} \max\{x, 0\}$, we have,

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

Proof: Squaring both sides of Eqn. (36), we have

$$\begin{aligned} &Q^2(t+1) \\ &\leq \big((Q(t) - \mu(t))^+\big)^2 + A^2(t) + 2\ A(t)(Q(t) - \mu(t))^+ \\ &\leq (Q(t) - \mu(t))^2 + A^2(t) + 2\ A(t)Q(t), \end{aligned}$$

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 $j \neq \mathtt{r}$,

$$\begin{aligned} &X_j^2(t+1) - X_j^2(t) \\ &\quad \leq Y(t) + 2X_j(t)\big(\sum_{m \in V} \mu_{mi_t^*}(t) - \sum_{k \in V} \mu_{kj}(t)\big), \end{aligned} \tag{37}$$

where $Y(t) \stackrel{\text{def}}{=} (\sum_{m \in V} \mu_{mi_t^*}(t))^2 + (\sum_{k \in V}\ \mu_{kj}(t))^2$. Let $C = \sum_e c_e$, the sum of the capacities of all links in the network. Now the node i_t^* could be the source node $\mathtt{r}$ or a non-source node in the network. In either case, since $\mu_e(t) \leq c_e$, $\forall e \in E$, the first term in $Y(t)$ above is upper-bounded by $\max\{A^2(t), C^2\}$ and the second term is upper-bounded by C^2. Hence, $Y(t) \leq \max\{A^2(t), C^2\} + C^2 \leq A^2(t) + 2C^2$. Since the number of arrivals per slot $A(t)$ is assumed to have bounded second moment, there exists a finite constant $B > 0$ such that $\mathbb{E}[Y(t)] \leq \mathbb{E}\big(A^2(t)\big) + 2C^2 \leq B$.

Now define a Quadratic Lyapunov function $L(\boldsymbol{X}(t)) \stackrel{\text{def}}{=} \sum_{j \neq \mathtt{r}} X_j^2(t)$. From Eqn. (37), the one-slot drift $\Delta(\boldsymbol{X}(t))$ of $L(\boldsymbol{X}(t))$ may be computed to be

$$\begin{aligned} &\Delta(\boldsymbol{X}(t)) \triangleq \mathbb{E}[L(\boldsymbol{X}(t+1)) - L(\boldsymbol{X}(t)) \mid \boldsymbol{X}(t)] \\ &= \mathbb{E}\big[\sum_{j \neq r} \big(X_j^2(t+1) - X_j^2(t)\big) \mid \boldsymbol{X}(t)\big] \\ &\leq B|V| + 2\sum_{j \neq r} X_j(t) \mathbb{E}\big[\sum_{m \in V} \mu_{mi_t^*}(t) - \sum_{k \in V}\ \mu_{kj}(t) \mid \boldsymbol{X}(t)\big] \\ &\stackrel{(a)}{=} B|V| - 2\sum_{(i,j) \in E} \mathbb{E}[\mu_{ij}(t) \mid \boldsymbol{X}(t)]\big(X_j(t) - \sum_{k \in K_j(t)} X_k(t)\big) \\ &= B|V| - 2\sum_{(i,j) \in E} \mathbb{E}[\mu_{ij}(t) \mid \boldsymbol{X}(t)]\, W_{ij}(t), \end{aligned} \tag{38}$$

where (a) follows from changing the order of summation and $K_j(t)$ and $W_{ij}(t)$ are as defined in Eqn. (15) and (16), respectively. To emphasize the fact that the drift upper-bound (38) depends on the control policy $\pi \in \Pi^*$, we attach a superscript π to the control variables $\boldsymbol{\mu}(t)$ as follows:

$$\Delta^\pi(\boldsymbol{X}(t)) \leq B|V| - 2\sum_{(i,j) \in E} \mathbb{E}[\mu_{ij}^\pi(t) \mid \boldsymbol{X}(t)]\, W_{ij}(t). \tag{39}$$

Our optimal broadcast policy $\pi^* \in \Pi^*$ is chosen to minimize the upper-bound on the drift expression, given by the right-hand side of Eqn. (39), among all policies in the space Π^*.

(b) Construction of a Stationary Randomized Policy π^{RAND} : Next, we construct an auxiliary randomized link-activation policy π^{RAND}, which will be useful later in the proof. Let the vector $\boldsymbol{\beta}^* \in \text{conv}\,(\mathcal{S})$ attain the upper-bound in Eqn. (5):

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

From Caratheodory's theorem [26], there exist at most $(|E| + 1)$ link-activation vectors $\{\boldsymbol{s}_k \in \mathcal{S}\}$ and associated non-negative scalars $\{p_k \geq 0\}$ with $\sum_{k=1}^{|E|+1} p_k = 1$, such that

$$\boldsymbol{\beta}^* = \sum_{k=1}^{|E|+1} p_k \boldsymbol{s}_k. \tag{40}$$

Hence, from Theorem 1 we have,

$$\lambda^* \leq \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e^*. \tag{41}$$

Consider an exogenous packet arrival rate λ at the source, which is strictly less than the broadcast capacity λ^*. Thus, there exists an $\epsilon > 0$ such that $\lambda + \epsilon \leq \lambda^*$. From Eqn. (41),

$$\lambda + \epsilon \leq \min_{U: \text{ a proper cut}} \sum_{e \in E_U} c_e \beta_e^*. \tag{42}$$

For any node $v \neq \mathtt{r}$ other than the source, consider the specific proper cuts $U_v = V \setminus \{v\}$, defined earlier in Eqn. (6). From Eqn. (42), we have

$$\lambda + \epsilon \leq \sum_{e \in E_{U_v}} c_e \beta_e^*, \quad \forall v \neq \mathtt{r}. \tag{43}$$

Since the underlying network topology $\mathcal{G} = (V, E)$ is a DAG, there exists a topological ordering of the nodes such that: (i) the nodes can be labelled serially as $\{v_1, \ldots, v_{|V|}\}$, where $v_1 = \mathtt{r}$ is the source node with no in-neighbours and the node $v_{|V|}$ has no outgoing neighbours and (ii) all edges in E are oriented from $v_i \to v_j$, $i < j$ [27]; From Eqn. (43), we define probabilities $q_j \in [0,1]$ for each node v_j such that

$$q_j \sum_{e \in E_{U_{v_j}}} c_e \beta_e^* = \lambda + \epsilon \frac{j}{|V|}, \quad j = 2, \ldots, |V|. \tag{44}$$

Consider a randomized link-activation policy π^{RAND} defined as follows: at every slot t (i) it randomly *selects* a feasible link-activation vector $\boldsymbol{s}(t) = \boldsymbol{s}_k$ with probability p_k, given in Eqn. (40), $k = 1, 2, \ldots, |E|+1$; (ii) for each selected link $e = (v_i, v_j)$, incoming to the node v_j with $s_e(t) = 1$, the link e is activated independently with probability q_j, given by Eqn. (44). The 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 node $j \in V$ is relabeled by the topological ordering as $v_l \in V$ for some $2 \le l \le |V|$, from Eqn. (44) we conclude that, for each node $j \neq \mathtt{r}$, the total expected incoming transmission rate to node j is given by

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

$$= q_l \sum_{e \in E_{U_{v_l}}} c_e \beta_e^* = \lambda + \epsilon \frac{l}{|V|}. \tag{45}$$

Equation (45) shows that under the randomized policy π^{RAND}, the total expected incoming capacity to each node $j \neq \mathtt{r}$ is strictly larger than the packet arrival rate λ. According to the abuse of notation in (14), at the source node $\mathtt{r}$ we have

$$\sum_{i:(i,\mathtt{r})\in E} \mathbb{E}[\mu_{i\mathtt{r}}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)] = \mathbb{E}[\sum_{i:(i,\mathtt{r})\in E} \mu_{i\mathtt{r}}^{\pi^{\text{RAND}}}(t)] = \lambda. \tag{46}$$

From Eqns. (45) and (46), if node i appears prior to node j in the aforementioned topological ordering, i.e., if $i \equiv v_{l_i} < v_{l_j} \equiv j$ for some $l_i < l_j$, then

$$\sum_{k:(k,i)\in E} \mathbb{E}[\mu_{ki}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)] - \sum_{k:(k,j)\in E} \mathbb{E}[\mu_{kj}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)] \le -\frac{\epsilon}{|V|} \tag{47}$$

(c) Stochastic Stability of $\{\boldsymbol{X}(t)\}_{t\ge 0}$ *under* π^* : The drift inequality (39) holds for any policy $\pi \in \Pi^*$. Our broadcast policy π^* observes the system state $\boldsymbol{X}(t)$ and minimizes the upper-bound on drift at every slot. Comparing the activations selected by the policy π^* with π^{RAND} in slot t, we have

$$\begin{aligned}\Delta^{\pi^*}(\boldsymbol{X}(t)) &\le B|V| - 2\sum\nolimits_{(i,j)\in E} \mathbb{E}\big[\mu_{ij}^{\pi^*}(t) \mid \boldsymbol{X}(t)\big]W_{ij}(t) \\ &\le B|V| - 2\sum_{(i,j)\in E} \mathbb{E}\big[\mu_{ij}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)\big]W_{ij}(t) \\ &= B|V| + 2\sum_{j\neq \mathtt{r}} X_j(t)\Big(\sum_{m\in V} \mathbb{E}\big[\mu_{mi_t^*}^{\pi^{\text{RAND}}}(t) \mid \boldsymbol{X}(t)\big] \\ &\quad - \sum_{k\in V} \mathbb{E}\big[\mu_{kj}^{\pi^{\text{RAND}}}(t) | \boldsymbol{X}(t)\big]\Big) \\ &\le B|V| - \frac{2\epsilon}{|V|}\sum_{j\neq r} X_j(t).\end{aligned} \tag{48}$$

Since node i_t^* is an in-neighbour of node j (10), the node i_t^* must appear before j in any topological ordering of the DAG $\mathcal{G}$. Hence, the inequality in (48) follows directly from (47). Taking expectation of both sides in (48) with respect to $\boldsymbol{X}(t)$,

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

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

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

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

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

Taking a $\limsup$ of both sides as $T \to \infty$, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{j\neq \mathtt{r}} \mathbb{E}[X_j(t)] \le \frac{B|V|^2}{2\epsilon}, \tag{49}$$

which implies that the virtual-queue process $\{\boldsymbol{X}(t)\}_{t=0}^{\infty}$ is strongly stable [8] under the policy $\pi^* \in \Pi^*$.

(d) Throughput-optimality of π^* : Finally, we show that the strong stability of the virtual queues $X_j(t)$ implies that the policy π^* achieves the broadcast capacity λ^* in a DAG, i.e., for all arrival rates $\lambda < \lambda^*$, we have

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

Equation (14) shows that the virtual queues $X_j(t)$ have bounded departures (due to the bounded link capacities). Thus, strong stability of $X_j(t)$ implies that all virtual queues $X_j(t)$ are rate stable [8, Th. 2.8], i.e., $\lim_{T\to\infty} X_j(T)/T = 0$, w.p.1 for all j. Using union-bound, it follows that,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{j\neq \mathtt{r}} X_j(T) = 0, \quad \text{w.p. 1} \tag{50}$$

Now consider any node $j \neq \mathtt{r}$ in the network. We can construct a simple path $\sigma(\mathtt{r} \overset{\text{def}}{=} u_k \to u_{k-1} \ldots \to u_1 \overset{\text{def}}{=} j)$ from the source node $\mathtt{r}$ to the node j by running Algorithm 3 on the DAG $\mathcal{G}(V, E)$.

Algorithm 3 $\mathtt{r} \to j$ Path Construction Algorithm

Require: Graph $\mathcal{G}(V,E)$, node $j \in V$

$i \leftarrow 1$
$u_i \leftarrow j$
while $u_i \neq \mathtt{r}$ **do**
  $u_{i+1} \leftarrow \arg\min_{m\in\delta_{\text{in}}(u_i)} Q_{mu_i}(t)$;
  $i \leftarrow i+1$
end while

Algorithm 3 chooses the parent of a node u_i in the path σ as the one that has the least relative packet deficit as compared to u_i. 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 $\sigma(\mathtt{r} \to j)$. 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,

$$\begin{aligned} R_j(T) &\equiv R_{u_1}(T) \\ &= \sum_{i=1}^{k-1}\big(R_{u_i}(T) - R_{u_{i+1}}(T)\big) + R_{u_k}(T) \\ &\overset{(*)}{=} -\sum_{i=1}^{k-1} X_{u_i}(T) + R_{\mathtt{r}}(T) \\ &= -\sum_{i=1}^{k-1} X_{u_i}(T) + \sum_{t=0}^{T-1} A(t), \end{aligned} \tag{51}$$

where the 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 the bound $\sum_{i=1}^{k-1} X_{u_i}(t) \leq \sum_{j\neq \mathtt{r}} X_j(t)$ (since $X_j(t) \geq 0$) and Eqn. (51), we conclude that for every node $j \neq \mathtt{r}$,

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

Finally, using the Strong Law of Large Numbers for the arrival process $\{A(t)\}_{t\geq 0}$ and Eqn. (50), we conclude

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

This concludes the proof.

C. Proof of Lemma 3

We regard the DAG $\mathcal{G}$ as a wire line network in which all links can be activated simultaneously at a slot. Theorem 2 and Eqn. (8) show that the broadcast capacity of the network $\mathcal{G}$ is

$$\begin{aligned} \lambda^* = \lambda_{\text{DAG}} = \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e &= \min_{\{U_v, v\neq \mathtt{r}\}} \sum_{e\in E_{U_v}} c_e \\ &\overset{(*)}{=} \min_{v\in V\setminus\{\mathtt{r}\}} d_{\text{in}}(v), \end{aligned} \tag{52}$$

where the sets U_v and E_{U_v} are defined in Eqns. (6) and (3) respectively. The equality $(*)$ follows from the assumption that $c_e = 1, \forall e \in E$. Edmond's Theorem [9] states that the maximum number of disjoint spanning trees in the graph $\mathcal{G}$ is

$$k^* = \min_{U:\text{ a proper cut}} \sum_{e\in E_U} c_e. \tag{53}$$

Combining (52) and (53) completes the proof of the Lemma.

References

[1] A. Karam, L. Zhang, and A. Lakas, "An efficient broadcasting scheme in support of military ad hoc communications in battle field," in *Proc. IEEE 9th Int. Conf. Innov. Inf. Technol. (IIT)*, Mar. 2013, pp. 78–83.

[2] *Livestream*, accessed on Jun. 27, 2016. [Online]. Available: http://new.livestream.com/

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, 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," *IEEE Trans. Inf. Theory*, vol. 48, no. 10, pp. 2690–2708, Oct. 2002.

[5] B. Pavkovic, W.-J. Hwang, and F. Theoleyre, "Cluster-directed acyclic graph formation for IEEE 802.15.4 in multihop topologies," in *Proc. IEEE 5th Int. Conf. New Technol., Mobility Security (NTMS)*, May 2012, pp. 1–6.

[6] A. Federgruen and Y.-S. Zheng, "Optimal power-of-two replenishment strategies in capacitated general production/distribution networks," *Manage. Sci.*, vol. 39, no. 6, pp. 710–727, 1993.

[7] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[8] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[9] R. Rustin, *Combinatorial Algorithms*. New York, NY, USA: Algorithmics Press, 1973.

[10] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. (WCNC)*, vol. 2. Mar. 2003, pp. 1124–1130.

[11] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Netw.*, vol. 8, nos. 2–3, pp. 153–167, 2002.

[12] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, "Randomized decentralized broadcasting algorithms," in *Proc. 26th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, May 2007, pp. 1073–1081.

[13] D. Towsley and A. Twigg, "Rate-optimal decentralized broadcasting: The wireless case," in *Proc. ACITA*, 2008, pp. 323–333.

[14] 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 *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2013, pp. 814–818.

[15] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *Proc. 43rd Annu. Allerton Conf. Commun., Control, Comput.*, 2005, pp. 797–815.

[16] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

[17] M. Chrobak, L. Gasieniec, and W. Rytter, "A randomized algorithm for gossiping in radio networks," in *Computing and Combinatorics*. Berlin, Germany: Springer, 2001, pp. 483–492.

[18] S. Basagni, D. Bruschi, and I. Chlamtac, "A mobility-transparent deterministic broadcast mechanism for ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 799–807, Dec. 1999.

[19] C. Joo, X. Lin, and N. B. Shroff, "Greedy maximal matching: Performance limits for arbitrary network graphs under the node-exclusive interference model," *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2734–2744, Dec. 2009.

[20] D. B. West, *Introduction to Graph Theory*, vol. 2. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[22] A. Brzezinski, G. Zussman, and E. Modiano, "Distributed throughput maximization in wireless mesh networks via pre-partitioning," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1406–1419, Dec. 2008.

[23] L. X. Bui, S. Sanghavi, and R. Srikant, "Distributed link scheduling with constant overhead," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1467–1480, Oct. 2009.

[24] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, vol. 6. Belmont, MA, USA: Athena Scientific, 1997.

[25] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, vol. 24. Berlin, Germany: Springer, 2003.

[26] J. Matoušek, *Lectures on Discrete Geometry*, vol. 108. New York, NY, USA: Springer, 2002.

[27] S. Dasgupta, C. H. Papadimitriou, and U. Vazirani, *Algorithms*. New York, NY, USA: McGraw-Hill, 2006.

Abhishek Sinha received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2010, and the M.E. degree in electronics and telecommunication engineering from the Indian Institute of Science, Bangalore, India, in 2012. He is currently pursuing the degree with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology. His research interests include optimization, information theory, and network control. He is a recipient of the Jagadis Bose National Science Talent Search Scholarship.

Georgios Paschos received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2002, and the Ph.D. degree in wireless networks from the Department of Electronics and Communication Engineering, University of Patras (supervisor Prof. S. Kotsopoulos), Greece, in 2006. He spent two years with the team of Prof. E. Modiano, Massachusetts Institute of Technology. From 2008 to 2014, he was with the Center of Research and Technology Hellas—Informatics and Telematics Institute, Greece, working with Prof. L. Tassiulas. Since 2014, he has been a Principal Researcher with Huawei Technologies Company, Ltd., Paris, France, leading the Network Control and Resource Allocation Team. He also taught with the Department of Electrical and Computer Engineering, University of Thessaly, from 2009 to 2011, as an Adjunct Lecturer. From 2007 to 2008, he was an ERCIM Post-Doctoral Fellow with VTT, Finland, working on the team of Prof. Norros. He serves as an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING and a TPC Member of the IEEE INFOCOM. Two of his papers received the best paper award in GLOBECOM 2007 and IFIP Wireless Days 2009, respectively.

Chih-Ping Li received the B.S. degree from National Taiwan University in 2001, and the M.S. and Ph.D. degrees from the University of Southern California in 2005 and 2011, respectively, all in electrical engineering. He was a Post-Doctoral Associate with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, from 2011 to 2014. Since 2014, he has been with the Corporate Research and Development Department, Qualcomm Research, and works on 5G wireless system design and analysis. His research interests include stochastic control, resource allocation, and QoS in communication networks, wireless networks, and queuing systems.

Eytan Modiano received the B.S. degree in electrical engineering and computer science from the University of Connecticut,CT, USA, Storrs, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, USA, in 1989 and 1992, respectively. He was a Naval Research Laboratory Fellow from 1987 to 1992, and a National Research Council Post-Doctoral Fellow from 1992 to 1993. From 1993 to 1999, he was with the Lincoln Laboratory, Massachusetts Institute of Technology (MIT). Since 1999, he has been a Faculty Member with MIT, where he is currently a Professor with the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems. His research is on communication networks and protocols with an emphasis on satellite, wireless, and optical networks. He was a co-recipient of the Sigmetrics 2006 Best Paper Award and the WiOpt 2013 Best Paper Award. He is an Editor-at-Large of the IEEE/ACM TRANSACTIONS ON NETWORKING, and served as an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE/ACM TRANSACTIONS ON NETWORKING. He was the Technical Program Co-Chair of the IEEE WiOpt 2006, the IEEE INFOCOM 2007, ACM MobiHoc 2007, and DRCN 2015. He is a Fellow of the IEEE and an Associate Fellow of the AIAA, and served on the IEEE Fellows Committee.