Throughput-Optimal Multi-hop Broadcast Algorithms

Abhishek Sinha*, Georgios Paschos† and Eytan Modiano*
*Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA
†Mathematical and Algorithmic Sciences Lab France Research Center, Huawei Technologies Co., Ltd.
Email: *sinhaa@mit.edu, †georgios.paschos@huawei.com, *modiano@mit.edu

***Abstract*—We design throughput-optimal dynamic broadcast algorithms for multi-hop networks with arbitrary topologies. Most of the previous broadcast algorithms route packets along spanning trees. For large time-varying networks, computing and maintaining a set of spanning trees is not efficient, as the network-topology may change frequently. In this paper we design a class of dynamic algorithms which make simple packet by packet scheduling and routing decisions and hence, obviate the need for maintaining any global topological structures, such as spanning trees. Our algorithms may be conveniently understood as a non-trivial generalization of the familiar *back-pressure* algorithm for unicast traffic, which performs packet routing and scheduling based on queue lengths. However, in the broadcast setting, due to packet duplications, it is difficult to define appropriate queuing structures. We design and prove the optimality of a virtual queue based algorithm, where virtual queues are defined for subsets of nodes. We then propose a multi-class broadcast policy which combines the above scheduling algorithm with *in-class-in-order* packet forwarding, resulting in significant reduction in complexity. Finally, we evaluate the performance of the proposed algorithms via extensive numerical simulations.**

***Index Terms*—Broadcasting, Network Control, Queueing Theory**

I. Introduction

Multihop broadcast routing is a fundamental network functionality for efficiently disseminating packets from a source node to *all* other nodes in a network. In this process, a *broadcast policy* is used to decide how to duplicate packets, and how to forward the duplicates over the network. The efficiency of a broadcast policy is measured in terms of its throughput, i.e., the maximum achievable common rate of packet reception by all nodes in the network. Formally, the *broadcast problem* refers to the problem of finding a broadcast policy that maximizes broadcast throughput and hence achieves the broadcast capacity in any network.

Broadcasting is extensively used in a number of important and diverse applications. Examples include military communications using ad-hoc networks [2], information dissemination in vehicular networks [3], live media streaming [4] and file searching [5], interactive video-on-demand service [6] and communication among multiple processors [7].

Solving the broadcast problem is challenging, especially for mobile wireless networks with time-varying connectivity. In this paper, we focus on designing dynamic broadcast algorithms. Such algorithms are robust with respect to the change of network topology, packet arrival rate and link quality. We derive a provably throughput-optimal dynamic broadcast algorithm for networks with arbitrary topology.

A. Related Work

The concept of broadcasting appears in many contexts. To avoid any confusion, it is important to distinguish at the outset *broadcast routing* (considered in this paper) where *a stream of packets is delivered to all nodes in a network*, from *broadcast transmissions* of an omnidirectional antenna, where packets are simultaneously transmitted to multiple wireless receivers in a single hop. The opposite extreme of broadcast routing is unicast, where a stream of packets is routed to a single destination node. An intermediate case of the above two scenarios is called *multicast*, where a stream of packets is to be replicated in a subset of nodes.

Most of the known throughput-optimal broadcast policies are static in nature and operate by forwarding copies of packets along pre-computed spanning trees [8]. In a network with time-varying topology, these static policies need to re-compute the trees every time the network topology changes, which is quite cumbersome and inefficient. Additionally, in most graphs, the number of possible spanning trees grows exponentially with the number of nodes in the graph. Our interest, therefore, is to design broadcast policies that are adaptive and do not require tree enumeration or mainteance.

A line of work from the domain of parallel computing has developed throughput optimal policies for complete graphs (cliques) connecting several processors [7], [9]. Other recent works [10], [11], [12] consider broadcasting on networks whose topology is a directed acyclic graph (DAG). In these works, broadcasting policies exploit the underlying network structure (either clique or DAG). However, these policies do not generalize to networks with arbitrary topologies.

The authors in [13] propose a randomized packet-forwarding policy for wireline networks, which is shown to be throughput-optimal under some assumptions. However, their policy potentially needs to use an unbounded amount of memory and can not be used in wireless networks with activation constraints.

A preliminary version of this paper appeared in the proceedings of MobiHoc, 2016, ACM [1].

This work was sponsored by NSF Grants CNS-1217048 and CNS-1524317.

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

A straight-forward extension of their policy, proposed in [14], uses an activation oracle, which is not practically feasible.

B. Our Contribution

In this paper, we address the throughput optimal broadcasting problem in arbitrary network topologies. Our main technical contributions are as follows:

(1) We first identify a convenient state-space representation of the network dynamics, in which the broadcast problem reduces to a "virtual-queue" stability problem, with appropriately defined virtual queues. By utilizing Stochastic Lyapunov-drift techniques, we derive a broadcast policy that provably achieves the broadcast capacity in arbitrary networks.

(2) Next, we introduce a multi-class heuristic policy, by combining the above policy with *in-class in-order* packet delivery from [10] in a suitable way. In this scheme, the number of classes is a tunable parameter, which offers a trade-off between efficiency and complexity. Several interesting properties of this heuristic scheme are also derived.

(3) Finally, we validate the theoretical ideas through extensive numerical simulations.

The rest of the paper is organized as follows. In Section II we describe the operational network model and characterize its broadcast capacity. In Section III we derive our throughput-optimal broadcast policy. In Section IV we propose a multi-class heuristic policy which uses the scheduling scheme derived in Section III. Section V describes the extension of the policy to wireless networks, while Section VI discusses distributed implementation of the proposed policy. In Section VII we validate our theoretical results via numerical simulations. Finally, in section VIII we conclude the paper with some directions for future work.

II. System Model

We begin our study with the consideration of broadcasting in wired networks with edge capacity constraints. This model is simple to describe and analytically tractable, yet it preserves the essential ingredients of the problem. The extension of the proposed broadcasting policy to wireless networks with activation constraints will be considered in Section V.

A. Network Model

Consider a graph $\mathcal{G}(V, E)$, V being the set of vertices and E being the set of directed edges, with $|V| = n$ and $|E| = m$. Time is slotted and the transmission capacity of each edge is one packet per slot. External packets arrive at the source node $\mathtt{r} \in V$. The arrivals are i.i.d. at every slot with an expected arrival of λ packets per slot.

To simplify the analysis, we perturb the slotted-time assumption and adopt a slightly different but equivalent *mini-slot* model. A slot consists of m consecutive mini-slots. Our dynamic broadcast algorithms are conceptually easier to derive, analyze and understand in the mini-slot model. However, the resulting algorithms can be easily adapted to the usual slotted model.

Mini-slot model: In this model, the basic unit of time is called a *mini-slot*. At each mini-slot t, an edge $e = (a, b) \in E$ is chosen for activation, independently and uniformly at random from the set of all m edges. All other $m-1$ edges remain idle for that mini-slot. A packet can be transmitted over an active edge only. A single packet transmission takes one mini-slot for completion. This random edge-activity process is represented by the i.i.d. sequence of random variables $\{S(t)\}_{t=1}^{\infty}$, such that, $S(t) = e$ indicates that the edge $e \in E$ is activated at the mini-slot t. Thus,

$$\mathbb{P}(S(t) = e) = 1/m, \quad \forall e \in E, \ \forall t$$

External packets arrive at the source $\mathtt{r}$ with expected arrival of λ/m packets per mini-slot.

The main analytical advantage of the mini-slot model is that only a single packet transmission takes place at a mini-slot, which makes it easier to express the system-dynamics. Moreover, we will show in Theorem (1) that the broadcast capacity is the same in the two models.

B. Broadcast-Capacity of a Network

Informally, a network supports a broadcast rate λ if there exists a scheduling policy, under which all nodes in the network receive packets at the rate of λ, for the same rate of packet arrival at the source. The broadcast-capacity λ^* is the maximally achievable broadcast rate in the network.

In the minislot model, we consider a class Π of scheduling policies, which observe the currently active edge $e = (a, b)$ at every mini-slot t and select at most one packet from node a and transmit it to b over the active edge e. On the other hand, in the slotted time model, admissible policies in Π may transmit at most one packet per edge *simultaneously* across all edges in the network at every slot. The policy-class Π includes policies that have access to all past and future information and may forward any packet present at node a at time t to node b.

Recall that, a slot corresponds to m consecutive mini-slots. In either model, let $R^\pi(T)$ be the number of distinct packets received in common by all nodes in the network, up to slot T, under a policy $\pi \in \Pi$. The time average $\lim_{T\to\infty} R^\pi(T)/T$ is the rate at which packets are received uniformly at all nodes.

Definition 1. *A policy $\pi \in \Pi$ achieves a broadcast throughput λ, if for a packet arrival rate of λ, we have*

$$\lim_{T\to\infty} \frac{1}{T} R^\pi(T) = \lambda, \quad \text{in probability.} \tag{1}$$

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

A policy, that achieves any rate $\lambda < \lambda^*$, is called a throughput-optimal policy. In the slotted-time model, the broadcast capacity λ^* of a network $\mathcal{G}$ follows from the Edmonds' tree-packing theorem [15], and is given by the following:

$$\lambda^* = \min_{\mathtt{t} \in V\setminus\{\mathtt{r}\}} \text{Max-Flow}(\mathtt{r} \to \mathtt{t}) \quad \text{per slot,} \tag{2}$$

where Max-Flow($\mathtt{r} \to \mathtt{t}$) denotes the maximum value of *flow* that can be feasibly sent from the node $\mathtt{r}$ to the node $\mathtt{t}$ in the graph $\mathcal{G}(V, E)$ [16]. Edmonds' tree-packing theorem also implies that there exist λ^* edge-disjoint arborescences [1] or directed spanning trees, rooted at $\mathtt{r}$ in the graph $\mathcal{G}$. By examining the flow from the source to every node and using (2), it follows that by sending unit flow over each edge-disjoint tree, we may achieve the capacity λ^*.
As an illustration, consider the graph shown in Figure 1. It follows from Eqn. (2) that the broadcast capacity of the graph is $\lambda^* = 2$. Edges belonging to a set of two edge-disjoint spanning trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are shown in blue and red in the figure.
The following theorem establishes the equivalence of the *mini-slot* model and the *slotted-time* model in terms of broadcast capacity.

Theorem 1 (Invariance of Capacity). *The broadcast capacity of the mini-slot model is the same as that of the slotted-time model and is given by Eqn.* (2).

Proof: See Appendix IX-A. ∎

III. A THROUGHPUT-OPTIMAL BROADCAST POLICY π^*

In this section, we design a throughput-optimal broadcast policy $\pi^* \in \Pi$, for networks with arbitrary topology. This algorithm is of *Max-weight* type and is inspired by the seminal back-pressure policy for the corresponding unicast problem [17]. However, because of packet duplications, the usual per-node queues cannot be defined here. We get around this difficulty by defining certain virtual-queues, corresponding to subsets of nodes. We show that a scheduling policy in Π, which *stochastically stabilizes* these virtual queues for all arrival rates $\lambda < \lambda^*$, constitutes a throughput-optimal broadcast policy. Based on this result, we derive a Max-Weight policy π^*, by minimizing the drift of a quadratic Lyapunov function of the virtual queues.

A. Definitions and Notations

To facilitate the description of our proposed algorithm, we first introduce the notion of *reachable sets* and *reachable sequence of sets* as follows:

Definition 3 (Reachable Set). *A subset of vertices $F \subset V$ is said to be* reachable *if the induced graph* [2] *$F(\mathcal{G})$ contains a directed arborescence, rooted at source $\mathtt{r}$, which spans the node set F.*

Equivalently, a subset of vertices $F \subset V$ is reachable if and only if there is a broadcast policy under which a packet p can be duplicated exactly in the subset F, during its course of broadcast. Note that, the set of all reachable sets is a strict subset of the set of all subsets of vertices. This is true because any reachable set, by definition, must contain the source node $\mathtt{r}$.

We may completely specify the *trajectory* of a packet during its course of broadcast, using the notion of *Reachable Sequences*, defined as follows:

Definition 4 (Reachable Sequence). *An ordered sequence of $n-1$ (reachable set, edge) tuples $\{(F_j, e_j), j = 1, 2, \ldots, n-1\}$ is called a* Reachable Sequence *if the following properties hold:*

- *$F_1 = \{r\}$ and for all $j = 1, 2, \ldots, n-1$:*
- *$F_j \subset F_{j+1}$*
- *$|F_{j+1}| = |F_j| + 1$.*
- *$e_j = (a, b) \in E : a \in F_j, b \in F_{j+1} \setminus F_j$*

$\mathcal{F}$ is defined to be the set of all reachable sequences.

A reachable sequence denotes a feasible sequence of transmissions for broadcasting a particular packet to all nodes, where the j^{th} transmission of a packet takes place across the edge $e_j, j = 1, 2 \ldots, n-1$. By definition, every reachable set must belong to at least one reachable sequence. A trivial upper bound on $|\mathcal{F}|$ is n^{2n}. An example illustrating the notions of reachable sets and reachable sequences for a simple graph is provided next.

Example: Consider the graph shown in Figure 1. A reachable sequence for this graph is given by $\mathcal{S}$ below:

$$\mathcal{S} = \{(\{\mathtt{r}\}, \mathtt{rc}), (\{\mathtt{r}, \mathtt{c}\}, \mathtt{ca}), (\{\mathtt{r}, \mathtt{a}, \mathtt{c}\}, \mathtt{rb})\}$$

This reachable sequence is obtained by adding nodes along the tree with red edges in Figure 1. Clearly, an example of a reachable set F in this graph is

$$F = \{\mathtt{r}, \mathtt{a}, \mathtt{c}\}$$

For a reachable set F, define its set of out-edges $\partial^+ F$ and in-edges $\partial^- F$ as follows:

$$\partial^+ F \equiv \{(a, b) \in E : a \in F, b \notin F\} \tag{3}$$

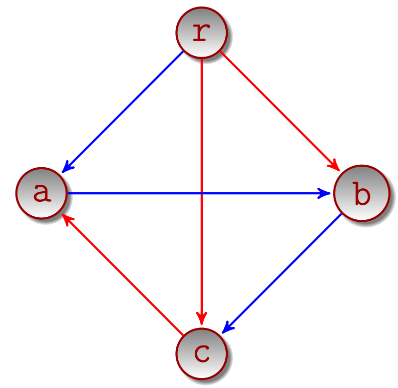

Fig. 1: The four-node diamond network $\mathcal{D}_4$.

[1] An *arborescence* is a directed graph such that there is a unique directed path from the root to all other vertices in it. Thus, an arborescence is a directed spanning tree. From now onwards, the terms "arborescence" and "directed spanning tree" will be used interchangeably.

[2] For a graph $\mathcal{G}(V, E)$ and any vertex set $F \subset V$, the induced graph $F(\mathcal{G})$ is defined as the sub-graph containing only the vertices F with the edges whose both ends lie in the set F.

$$\partial^- F \equiv \{(a,b) \in E : a \in F, b \in F\} \quad (4)$$

For an edge $e = (a,b) \in \partial^+ F$, define

$$F + e \equiv F \cup \{b\} \quad (5)$$

Similarly, for an edge $e = (a,b) \in \partial^- F$, define

$$F \setminus \{e\} \equiv F \setminus \{b\} \quad (6)$$

Convergence of Random Variables: For a sequence of random variables $\{X_n\}_{n=1}^\infty$ and another random variable X, defined on the same probability space, by the notation $X_n \stackrel{p}{\Longrightarrow} X$ we mean that the sequence of random variables $\{X_n\}_{n=1}^\infty$ converges *in probability* to the random variable X [18].

B. System Dynamics

Consider any broadcast policy $\pi \in \Pi$ in action. For any reachable set $F \subsetneq V$, denote the number of packets, replicated *exactly* at the vertex-set F at mini-slot t, by $Q_F(t)$[3]. A packet p, which is replicated exactly at the set F by time t, is called a *class-F packet*. Hence, at a given time t, the reachable sets $F \in \mathcal{F}$ induce a disjoint partition of all the packets present in the network.

In our mini-slot model, a class-F packet can make a transition only to class $F+e$ (where $e \in \partial^+ F$) during a mini-slot, where e is the active edge. Let the rate allocated to the edge e, for transmitting a class-F packet at time t, be denoted by $\mu_{e,F}(t)$ (naturally, $\mu_{e,F}(t) \equiv 0$, if F is not a reachable set or e is inactive) [4]. Here $\mu_{e,F}(t)$ is a binary-valued control variable, which assumes the value 1 if the active edge e is allocated to transmit a class-F packet at the mini-slot t.

In the following we argue that, for any reachable set F, the variable $Q_F(t)$ satisfies the following one-step queuing-dynamics (Lindley recursion) [19]:

$$\begin{aligned} Q_F(t+1) &\leq \Big(Q_F(t) - \sum_{e \in \partial^+ F} \mu_{e,F}(t)\Big)^+ + \quad (7) \\ &\quad \sum_{(e,G): e \in \partial^- F, G = F \setminus \{e\}} \mu_{e,G}(t), \quad \forall F \neq \{\mathtt{r}\} \\ Q_{\{\mathtt{r}\}}(t+1) &\leq \Big(Q_{\{\mathtt{r}\}}(t) - \sum_{e \in \partial^+(\{\mathtt{r}\})} \mu_{e,\{\mathtt{r}\}}(t)\Big)^+ + A(t) \end{aligned}$$

The dynamics in Eqn. (7) may be derived as follows: in the mini-slot model, only one packet over the currently active edge can be transmitted in the entire network at any mini-slot. Hence, for any reachable set F, the value of the corresponding state-variable $Q_F(t)$ may go up or down by at most one in a mini-slot. Now, $Q_F(t)$ decreases by one when any of the out-edges $e \in \partial^+ F$ is activated at mini-slot t and it carries a class-F packet, provided $Q_F(t) > 0$. This explains the first term in Eqn. (7). Similarly, the variable $Q_F(t)$ increases by one when a packet in some set $G = F \setminus \{e\}$ (or an external packet, in case $F = \{\mathtt{r}\}$), is transmitted to the set F over the (active) edge $e \in \partial^- F$. This explains the second term in Eqn. (7). In the following, we slightly abuse the notation by setting $\sum_{(e,G): e \in \partial^- F, G = F \setminus \{e\}} \mu_{e,G}(t) \equiv A(t)$, when $F = \{\mathtt{r}\}$. With this convention, the system dynamics is completely specified by the first inequality in (7), which constitutes a discrete time **Lindley recursion** [19].

[3] In the rest of the paper, we define $Q_V(t) = 0, \forall t$.

[4] Note that $\mu_{e,F}(t)$ and consequently, $Q_F(t)$ depend on the used policy π and should be denoted by $\mu^\pi_{e,F}(t)$ and $Q^\pi_F(t)$. Here we drop the superscript π to simplify notation.

C. Relationship between Stability and Throughput Optimality

The following lemma shows that stability of the virtual queues implies throughput-optimality for any admissible policy.

Lemma 1 (**Stability implies Throughput-Optimality**). *Consider a Markovian policy π, under which the induced Markov Chain $\{\boldsymbol{Q}^\pi(t)\}_0^\infty$ is Positive Recurrent for all arrival rate $\lambda < \lambda^*$. Then π is a throughput optimal broadcast policy.*

Proof: Under the action of a Markovian Policy π, the total number of packets $R^\pi(T)$ delivered to all nodes in the network by the time T is given by

$$R^\pi(T) = \sum_{t=1}^T A(t) - \sum_F Q^\pi_F(T)$$

Hence, the rate of packet broadcast is given by

$$\begin{aligned} \lim_{T\to\infty} \frac{R^\pi(T)}{T} &= \lim_{T\to\infty} \Big(\frac{1}{T}\sum_{t=1}^T A(t) - \sum_F \frac{Q^\pi_F(T)}{T}\Big) \\ &\stackrel{p}{\Longrightarrow} \lambda - \sum_F \lim_{T\to\infty} \frac{Q^\pi_F(T)}{T} \quad (8) \\ &\stackrel{p}{\Longrightarrow} \lambda \quad (9) \end{aligned}$$

Eqn. (8) follows from the Weak Law of Large Numbers for the arrival process. To justify Eqn. (9), note that for any $\delta > 0$ and any reachable set F, we have

$$\lim_{T\to\infty} \mathbb{P}\Big(\frac{Q^\pi_F(T)}{T} > \delta\Big) = \lim_{T\to\infty} \mathbb{P}\Big(Q^\pi_F(T) > T\delta\Big) = 0, \quad (10)$$

where the last equality follows from the assumption of positive recurrence of $\{\boldsymbol{Q}^\pi(t)\}$. Thus Eqn. (10) implies that $\frac{Q^\pi_F(T)}{T} \stackrel{p}{\Longrightarrow} 0, \forall F$. This justifies Eqn. (9) and proves the lemma. ■

1) Stochastic Stability of the Process $\{Q(t)\}_{t\geq 1}$: Equipped with Lemma (1), we now focus on finding a Markovian policy π^*, which stabilizes the chain $\{\boldsymbol{Q}^{\pi^*}(t)\}_{t\geq 1}$[5]. To accomplish this goal, we use the Lyapunov drift methodology [20], and derive a dynamic policy π^* which minimizes the one-minislot drift of a certain Lyapunov function. We then show that the proposed policy π^* has negative drift outside a bounded region in the state-space. Upon invoking the Foster-Lyapunov criterion [21], this proves positive recurrence of the chain $\{\boldsymbol{Q}(t)\}_0^\infty$.

[5] The argument t denotes time in mini-slots.

To apply the scheme outlined above, we start out by defining the following Quadratic Lyapunov Function $L(\boldsymbol{Q}(t))$:

$$L(\boldsymbol{Q}(t)) = \sum_{F} Q_F^2(t), \tag{11}$$

where the sum extends over all reachable sets. Recall that, the r.v. $S(t)$ denotes the currently active edge at the mini-slot t. The one-minislot drift is defined as:

$$\Delta_t(\boldsymbol{Q}(t), S(t)) \equiv L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) \tag{12}$$

From the dynamics (7), we have

$$\begin{aligned} &Q_F^2(t+1) \leq Q_F^2(t) + \mu_{\max}^2 \\ -\quad &2Q_F(t)\bigg(\sum_{e\in\partial^+F}\mu_{e,F}(t) - \sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\mu_{e,G}(t)\bigg), \end{aligned}$$

where $\mu_{\max} = 1$ is the maximum capacity of a link per mini-slot. Thus, one mini-slot drift may be upper-bounded as follows:

$$\begin{aligned} &\Delta_t(\boldsymbol{Q}(t), S(t)) \leq 2^n \mu_{\max}^2 \\ &-2\sum_{F\subsetneq V} Q_F(t)\bigg(\sum_{e\in\partial^+F}\mu_{e,F}(t) - \sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\mu_{e,G}(t)\bigg). \end{aligned}$$

Interchanging the order of summation, we have

$$\begin{aligned} \Delta_t(\boldsymbol{Q}(t), S(t)) \quad \leq \quad & 2^n \mu_{\max}^2 \\ & - \sum_{(e,F):e\in\partial^+F}\mu_{e,F}(t)\Big(Q_F(t) - Q_{F+e}(t)\Big). \end{aligned}$$

Taking expectation of both sides of the above inequality with respect to the edge-activation process $S(t)$ and the arrival process $A(t)$, we obtain the following upper-bound on the conditional Lyapunov drift $\Delta_t(\boldsymbol{Q}(t))$:

$$\begin{aligned} &\Delta_t(\boldsymbol{Q}(t)) \equiv \mathbb{E}_{S(t)}\Delta_t(\boldsymbol{Q}(t), S(t)) \qquad (13)\\ &\leq 2^n\mu_{\max}^2 - \\ &\sum_{(e,F):e\in\partial^+F}\Big(Q_F(t) - Q_{F+e}(t)\Big)\mathbb{E}\big(\mu_{e,F}(t)|\boldsymbol{Q}(t), S(t)\big). \end{aligned}$$

Due to the activity constraint, if $S(t) = e$, we must have $\mu_{l,G}(t) = 0, \forall l \neq e$, for all reachable sets G. In other words, a packet can only be transmitted along the *active* edge for the mini-slot t.

For any reachable set F with an out-edge $e \in \partial^+F$, define the weight

$$w_{F,e}(t) = Q_F(t) - Q_{F+e}(t). \tag{14}$$

Consider the following Max-weight policy π^*, which transmits a packet p^* belonging to class-F from node i, where the packet p^* has the highest positive weight $w_{F,e}^*(t) = \max_F w_{F,e}(t)$, from the set of all packets contending for the edge e at mini-slot t. The resulting policy is presented formally in Algorithm 1.

Algorithm 1 The Dynamic Broadcast Policy π^*

1: Select an edge e for activation independently and uniformly at random from the set of all edges E.
2: Compute all reachable sets F such that $e \in \partial^+F$.
3: Transmit a class-F packet over the edge e, such that the corresponding weight $w_{F,e}(t) = Q_F(t) - Q_{F+e}(t)$ is positive and achieves the maximum over all such reachable sets F, computed in step 1 above. (Recall, $Q_V(t) = 0, \forall t$).
4: Idle, if no such F exists.

We now state the main theorem of this paper.

Theorem 2 (Throughput-Optimality of π^*)**.** *The dynamic policy π^* is a throughput-optimal broadcast policy for any network.*

Proof: See Appendix (IX-B). ■

Discussion: A straightforward way to extend the resulting policy to the slotted-time model (where all edges can simultaneously transmit packets at every slot) would be to transmit a packet p_e from the class $F_e^* = \arg\max_{F:e\in\partial^+F} w_{F,e}(t)$ over the edge $e, \forall e \in E$. Note that, the weights $w_{F,e}(t)$ are computed based on the queue-lengths $Q_F(t)$ at the beginning of slot t.

Note that, the policy π^* makes dynamic routing and scheduling decision for each packet, based on the current network state vector $\boldsymbol{Q}(t)$. However, to implement the policy π^* exactly, the nodes need to keep track of global state information, which appears to be prohibitive. In the next section, we design a heuristic version of the policy π^*, which is decentralized and is conjectured to be throughput-optimal based on extensive simulation results.

IV. A Multi-Class Broadcasting Heuristic

A potential difficulty in implementing the policy π^* is that one needs to maintain a state-variable $Q_F(t)$, corresponding to each reachable set F, and keep track of the particular reachable set $F_p(t)$, to which packet p belongs. For large networks, without any additional structure in the scheduling policy, maintaining such a detailed state information is quite cumbersome. To alleviate this problem, we next propose a heuristic policy which combines π^* with the idea of *in-class in-order delivery*. The introduction of class-based in-order delivery imposes additional structure in the packet scheduling, which in turn, substantially reduces the complexity of the state-space.

Motivation: To motivate the heuristic policy, we begin with a simple policy space $\Pi^{\text{in-order}}$, first introduced in [10] for throughput-optimal broadcasting in wireless Directed Acyclic Graphs (DAG). Policies in $\Pi^{\text{in-order}}$ deliver packets to nodes according to their order of arrival at the source. Unfortunately, as shown in [10], although $\Pi^{\text{in-order}}$ is sufficient for achieving throughput-optimality in a DAG, it is not necessarily throughput-optimal for arbitrary networks, containing directed cycles. To tackle this problem, we generalize

the idea of in-order delivery by proposing a k-class policy space $\Pi_k^{\text{in-order}}, k \geq 1$, which generalizes the space $\Pi^{\text{in-order}}$. In this policy-space, the policies divide the packets into k distinct classes. The in-order delivery constraint is imposed within each class but not across different classes. Thus, in $\Pi_k^{\text{in-order}}$, the scheduling constraint of $\Pi^{\text{in-order}}$ is relaxed by requiring that packets belonging to *each individual class* be delivered to nodes according to their order of arrival at the source. However, the space $\Pi_k^{\text{in-order}}$ does not impose any orderly requirement for deliveries of packets across different classes. Combining it with the max-weight scheduling scheme, designed earlier for the throughput-optimal policy π^*, we propose a multi-class heuristic policy $\pi_k^H \in \Pi_k^{\text{in-order}}$ which is *conjectured* to be throughput-optimal for large-enough number of classes k. Extensive numerical simulations have been carried out to support this conjecture.
The following section gives a detailed description of this heuristic policy, outlined above.

A. The In-order Policy Space $\Pi^{\text{in-order}}$

Now we formally define the policy space $\Pi^{\text{in-order}}$:

Definition 5 (Policy-Space $\Pi^{\text{in-order}}$ [10])**.** *A broadcast policy π belongs to the space $\Pi^{\text{in-order}}$ if all incoming packets at the source r are serially indexed $\{1, 2, 3, \dots\}$ according to their order of arrivals, and a node $i \in V$ is allowed to receive a packet p at time t only if the node i has received the packets $\{1, 2, \dots, p-1\}$ by time t.*

As a result of the *in-order* delivery property of policies in the space $\Pi^{\text{in-order}}$, it follows that the state of received packets in the network at time t may be completely represented by the n-dimensional vector $\boldsymbol{R}(t)$, where $R_i(t)$ denotes the highest index of the packet received by node $i \in V$ by time t. We emphasize that this succinct representation of network state is valid only under the action of the policies in the space $\Pi^{\text{in-order}}$, and is not necessarily true in the general policy space Π.
Due to the highly-simplified state-space representation, it is natural to try to find efficient broadcast-policies in the space $\Pi^{\text{in-order}}$ for arbitrary network topologies. We showed in our earlier work [10] that if the underlying topology of the network is restricted to DAGs, the space $\Pi^{\text{in-order}}$ indeed contains a throughput-optimal broadcast policy. However, we also proved that the space $\Pi^{\text{in-order}}$ is not rich enough to achieve broadcast capacity in networks with arbitrary topology. We re-state the following proposition in this connection.

Proposition 3. (THROUGHPUT-LIMITATION OF THE POLICY SPACE $\Pi^{\text{in-order}}$ [10]) *There exists a network $\mathcal{G}$ such that, no broadcast-policy in the space $\Pi^{\text{in-order}}$ can achieve the broadcast-capacity of $\mathcal{G}$.*

The above proposition is proved in [10], by showing that no broadcast policy in the space $\Pi^{\text{in-order}}$ can achieve the broadcast-capacity in the diamond-network $\mathcal{D}_4$, depicted in Figure 1.

B. The Multi-class Policy-Space $\Pi_k^{\text{in-order}}$

To overcome the throughput-limitation of the space $\Pi^{\text{in-order}}$, we propose the following generalized policy space $\Pi_k^{\text{in-order}}, k \geq 1$, which retains the efficient representation property of the space $\Pi^{\text{in-order}}$.

Definition 6 (Policy-Space $\Pi_k^{\text{in-order}}$)**.** *A broadcast policy π belongs to the space $\Pi_k^{\text{in-order}}$ if the following conditions hold:*

- *There are k distinct "classes".*
- *A packet, upon arrival at the source, is labelled with any one of the k classes, uniformly at random. The label of a packet remains fixed throughout its course of broadcast.*
- *Packets belonging to each individual class $j \in [1, \dots, k]$, are serially indexed $\{1, 2, 3, \dots\}$ according to their order of arrival.*
- *A node $i \in V$ in the network is allowed to receive a packet p from class j at time t, only if the node i has received the packets $\{1, 2, \dots, p-1\}$ from the class j by time t .*

In other words, in the policy space $\Pi_k^{\text{in-order}}$, packets belonging to each individual class $j \in [1, \dots, k]$ are delivered to nodes *in-order*. It is also clear from the definition that

$$\Pi_1^{\text{in-order}} = \Pi^{\text{in-order}}$$

Thus, the collection of policy-spaces $\{\Pi_k^{\text{in-order}}, k \geq 1\}$ generalizes the policy space $\Pi^{\text{in-order}}$.

State-Space representation under $\Pi_k^{\text{in-order}}$: Since each class in the policy space $\Pi_k^{\text{in-order}}$ obeys the in-order delivery property, it follows that the network state at time t is completely described by the k-tuple of vectors $\{\boldsymbol{R}^c(t), 1 \leq c \leq k\}$, where $R_i^c(t)$ denotes the highest index of the packet received by node $i \in V$ from class c by time t. Thus the state-space complexity grows *linearly* with the number of classes used.
Following our development so far, it is natural to seek a throughput-optimal broadcast policy in the space $\Pi_k^{\text{in-order}}$ with a *small* class-size k. In contrast to Proposition (3), the following proposition gives a positive result in this direction.

Proposition 4. (THROUGHPUT-OPTIMALITY OF THE SPACE $\Pi_k^{\text{in-order}}, k \geq n/2$) *For every network $\mathcal{G}$, there exists a throughput-optimal broadcast policy in the policy space $\Pi_k^{\text{in-order}}$, for all $k \geq n/2$.*

The proof of this proposition uses a static policy, which routes the incoming packets along a set of λ^* edge-disjoint spanning trees. For a network with broadcast-capacity λ^*, the existence of these trees are guaranteed by Edmonds' tree packing theorem [15]. Then we show that for any network with unit-capacity edges, its broadcast-capacity λ^* is upper-bounded by $n/2$, which completes the proof. The details of this proof are outlined in Appendix IX-E.

C. General Properties of the Multi-class Policy-Space

In this subsection we show how the intra-class in-order delivery property of the multi-class policy space constrains the delivery of packets per class. In particular, we show that at any time the number of distinct subsets of nodes, where packets from any class belong to, is at most $n+1$. This should be contrasted with the unrestricted policy space π, where the packets at any time may be present in all subsets of nodes, which is exponential in the size of the network.

To formally state the property, define $F_p^{(j)}(t) \subseteq V$ to be the subset of nodes where the p^{th} packet from class j belongs to at time t. We claim that,

Proposition 5. *For any $1 \leq p_1 < p_2$ and for any time t, we have*

$$F_{p_2}^{(j)}(t) \subseteq F_{p_1}^{(j)}(t) \tag{15}$$

Proof: If $F_{p_2}^{(j)}(t) = \phi$, the inclusion holds trivially. Otherwise, consider a node $v \in F_{p_2}^{(j)}(t)$. This implies that the node v containts the ${p_2}^{\text{th}}$ packet from class k at time t. Since all classes in the policy space $\Pi_k^{\text{in-order}}$ satisfies the in-order delivery property, it follows that the node v must contain the ${p_1}^{\text{th}}$ packet from class k at time t, where $p_1 < p_2$. Thus $v \in F_{p_1}^{(j)}(t)$. This implies that $F_{p_2}^{(j)}(t) \subseteq F_{p_1}^{(j)}(t)$, which proves the proposition. ■

The above proposition immediately implies the following interesting result. Let $\mathcal{F}^{(j)}(t)$ denote the family of distinct subsets of nodes where packets from class k are present at time t, i.e.,

$$\mathcal{F}^{(j)}(t) = \{F_p^{(j)}(t) | p \geq 1\} \tag{16}$$

Proposition 6. *For all classes $1 \leq j \leq k$ and all time $t \geq 1$, we have*

$$|\mathcal{F}^{(j)}(t)| \leq n+1 \tag{17}$$

Proof: Using Proposition (5), we have the following chain of set inclusions

$$V \supseteq F_1^{(j)}(t) \supseteq F_2^{(j)}(t) \supseteq \ldots \supseteq F_p^{(j)}(t) \supseteq \ldots$$

Since $|V| = n$ and the sequence of sets of vertices $\{F_i^{(j)}(t)\}_{i \geq 1}$ are *decreasing*, there could be at most $n+1$ distinct sets in the family $\mathcal{F}^{(j)}(t)$. ■

Discussions: Proposition 6 suggests that each individual class is structurally constrained in disseminating packets. Without the in-order restriction, we trivially have $|\mathcal{F}^{(j)}(t)| = O(2^n)$. On the other hand, under the action of any broadcast policy which routes packet along a fixed spanning tree, it is easy to see that the statement of Eqn. (17) holds. The surprising conclusion of Proposition 6 is that it shows that the statement of Eqn. (17) holds good even when we do not restrict the individual classes to follow a fixed spanning tree, but require them to respect a much weaker assumption of *in-order* delivery only. As a consequence, it is natural to search for an efficient broadcast policy with multiple classes, so that, the packet-delivery restriction of each individual class may be overcome collectively.

D. A Multi-class Heuristic Policy $\pi_k^H \in \Pi_k^{\text{in-order}}$

Since any policy in the class $\Pi_k^{\text{in-order}}$ delivers packets from the same class *in-order*, the *intra-class* packet scheduling is fixed for the entire policy-class $\Pi_k^{\text{in-order}}$. Thus, we only need to specify an *inter-class* scheduling policy to resolve contentions among multiple packets from different classes to access an active edge for transmission. In this sub-section, we propose a dynamic policy $\pi_k^H \in \Pi_k^{\text{in-order}}$, which uses the same Max-Weight packet scheduling rule, as the throughput-optimal policy π^*, for *inter-class* packet scheduling. As we will see, the computation of weights and packet scheduling in this case may be efficiently carried out by exploiting the special structure of the space $\Pi_k^{\text{in-order}}$.

Motivation: We observe that, when the number of classes $k = \infty$, so that every incoming packet to the source $\mathtt{r}$ joins a new class, the *in-order* restriction of the space $\Pi_k^{\text{in-order}}$ is essentially no longer in effect. In particular, the throughput-optimal policy π^* of Section III belongs to the space $\Pi_\infty^{\text{in-order}}$. This motivates us to consider the following multi-class scheduling policy π_k^H:

Intra-class packet scheduling: Recall that, under a policy $\pi \in \Pi_k^{\text{in-order}}$, a packet arriving at the source $\mathtt{r}$, joins one of the k classes uniformly at random. Packets belonging to any class $c = 1, 2, \ldots, k$ are delivered to all nodes *in-order* (i.e. the order they arrived at the source $\mathtt{r}$). Let the state-variable $R_i^c(t)$ denote the number of packets belonging to the class c received by node i up to the mini-slot t, $i = 1, 2, \ldots, n$, $c = 1, 2, \ldots, k$. As discussed earlier, given the intra-class in-order delivery restriction, the state of the network at the mini-slot t is completely specified by the vector $\{\boldsymbol{R}^c(t), c = 1, 2, \ldots, k\}$. Due to the in-order packet-delivery constraint, when an edge $e = (i, j)$ is active at the mini-slot t, not all packets that are present at node i and not-present at node j are eligible for transmission. Under the policy $\pi_k^H \in \Pi_k^{\text{in-order}}$, only the next *Head-of-the-Line* (HOL) packet from each class, i.e., packet with index $R_j^c(t) + 1$ from the class c, $c = 1, 2, \ldots, k$ are eligible to be transmitted to the node j, provided that the corresponding packet is also present at node i by mini-slot t. Hence, at a given mini-slot t, there are at most k contending packets for an active edge. This should be compared with the policy π^*, in which there could be $\Theta(2^n)$ contending packets for an active edge at a mini-slot.

Inter-class packet scheduling: Given the above intra-class packet-scheduling rule, which follows directly from the definition of the policy space $\Pi_k^{\text{in-order}}$, we now propose an inter-class packet scheduling, for resolving the contention among multiple contending classes for an active edge e at a mini-slot t. For this purpose, we utilize the *same Max-Weight scheduling rule, derived for the policy* π^* (step 2 of Algorithm 1).

The main computational advantage of the multiclass policy

π_k^H over the throughput-optimal policy π^* is that, instead of computing the weights $w_{F,e}(t)$ in (14) for all reachable sets F, we only need to compute the weights of the sets F_c corresponding to the HOL packets (if any) belonging to the class c. By exploiting the structure of the space $\Pi_k^{\text{in-order}}$, this requires quadratic number of computations in the class-size k (see Algorithm 2) per mini-slot. Finally, we schedule the HOL packet from the class c^* having the maximum (positive) weight.

Keeping in mind our earlier discussion about similarity of packet forwarding capabilities of the classes and trees, we put forward the following conjecture regarding the performance of the proposed heuristic:

Conjecture 1. *The multiclass policy π_k^H is throughput-optimal for $k = \Theta(\lambda^*)$, where λ^* is the broadcast capacity of the network.*

Extensive numerical simulation results supporting the conjecture will be presented in Section VII-C.

Pseudo code: The full pseudo code of the policy π_k^H is provided in Algorithm 2. In lines $4 \ldots 10$, we have used the in-order delivery property of the policy π_k^H to compute the sets F_c, to which the next HOL packet from the class c belongs. This property is also used in computing the number of packets in the set $G = F_c, F_{c+e}$ in line 14 as follows: recall that, the variable $Q_G(t)$ counts the number of packets that the reachable set G contains exclusively at mini-slot t. These packets can be counted by counting such packets from each individual classes and then summing them up. Again utilizing the in-class in-order delivery property, we conclude that the number of packets $N_G^c(t)$ from class c, that belongs exclusively to the set G at time t is given by

$$N_G^c(t) = \Big(\min_{i \in G} R_i^c(t) - \max_{i \in V \setminus G} R_i^c(t)\Big)^+.$$

Hence,

$$Q_G(t) = \sum_{c=1}^{k} N_G^c(t),$$

which explains the assignment in line 19. In line 23, the *weights* corresponding to the HOL packets of each class are computed according to Eqn. (14). Finally, in line 25, the HOL packet with the highest positive weight is transmitted across the active edge e. The per mini-slot complexity of the policy π_k^H is $\mathcal{O}(nk)$.

Algorithm 2 The Multi-class Scheduling Policy π_k^H

At each mini-slot t, the network-controller observes the state-variables $\{R_l^c(t), l \in V, c = 1, 2, \ldots, k\}$, the currently active edge $S(t) = e = (i, j)$ and executes the following steps at node i:

1: **for all** classes $c = 1 : k$ **do**
2: */* Determine the index of the next expected **in-order** (HOL) packet p_c from the class c for node j */*
3: $p_c \leftarrow R_j^c(t) + 1$.
4: */* Check whether node i has more packets than node j belonging to class c */*
5: **if** $R_i^c(t) < p_c$ **then**
6: $w_c \leftarrow 0$
7: continue;
8: **end if**
9: */* Find the subset $F_c \subset V$ where the packet p_c is currently present */*
10: $F_c \leftarrow \phi$
11: **for all** node $l = 1 : n$ **do**
12: **if** $R_l^c(t) \geq p_c$ **then**
13: $F_c \leftarrow F_c \cup \{l\}$
14: **end if**
15: **end for**
16: $F_{c+e} = F_c \cup \{j\}$
17: */* Find $Q_{F_c}(t)$ and $Q_{F_c+e}(t)$ */*
18: **for** $G = F_c$ and F_{c+e} **do**
19: $Q_G(t) \leftarrow \sum_{c=1}^{k} \Big(\min_{i \in G} R_i^c(t) - \max_{i \in V \setminus G} R_i^c(t)\Big)^+$
20: **end for**
21: $Q_V(t) \leftarrow 0$
22: */* Compute the weight w_c for packet p_c */*
23: $w_c \leftarrow \big(Q_{F_c}(t) - Q_{F_c+e}(t)\big)$
24: **end for**
25: Schedule the packet $p^* \in \arg\max_c w_c$, when $\max_c w_c > 0$, else idle.

V. Broadcasting in Wireless Networks

A wireless network is modeled by a graph $\mathcal{G}(V, E)$, along with a set of edge-subsets $\mathcal{M}$ (represented by a set of binary characteristic vectors of dimension $|E| = m$). The set $\mathcal{M}$ is called the set of all *feasible activations* [17]. The structure of the set $\mathcal{M}$ depends on the underlying interference constraint, e.g., under the *primary interference constraint*, the set $\mathcal{M}$ consists of all *matchings* of the graph $\mathcal{G}$ [22]. Any subset of edges $\boldsymbol{s} \in \mathcal{M}$ can be activated simultaneously at a given slot. For broadcasting in wireless networks, we first activate a feasible subset of edges from $\mathcal{M}$ and then forward packets on the activated edges.

Since the proposed broadcast algorithms in sections III and IV are *Max-Weight* by nature, they extend straight-forwardly to wireless networks with activation constraints [20]. In particular, from Eqn. (14), at each slot t, we first compute the weight of each edge, defined as $w_e(t) = \max_{F: e \in \partial^+ F} w_{e,F}(t)$. Next, we activate the subset of edges $\boldsymbol{s}^*(t)$ from the activation set $\mathcal{M}$, having the maximum weight, i.e.,

$$\boldsymbol{s}^*(t) = \arg\max_{\boldsymbol{s} \in \mathcal{M}} \sum_{e \in E} w_e(t) s_e$$

Packet forwarding over the activated edges remains the same as before. The above activation procedure carries over to the multi-class heuristic π_k^H in wireless networks.

VI. Distributed Implementation

From the description of Algorithm 2, we note that the weight for a class c at a node i is computed based on the knowledge of the current HOL packet indices $\{R_j^c(t)\}$ of *all* nodes in the network. Gathering this global state information in a centralized fashion consumes precious network resources. To overcome this issue, we propose the following local message-passing algorithm for exchanging pairwise state information. Each node maintains an $n \times k$ state table consisting of the *last known* HOL packet indices $\hat{\boldsymbol{R}}(t) = \{\hat{R}_i^{(c)}(t)\}$, along with the timestamps $t_j(t)$ of when the corresponding entry was generated. Observe that, the entry $R_i^c(t)$ corresponding to node i is locally known to node i, and is always fresh. However, entries corresponding to other nodes $\{\hat{\boldsymbol{R}}_j(t), j \neq i\}$ may be outdated. The timestamp information is used to keep the locally known state-information at each node as fresh as possible. If an edge (i, j) is activated during a minislot, the nodes i and j exchange their state table w.p. q, where $0 < q \leq 1$ is a tunable parameter. The entry $\boldsymbol{R}_k(t)$ corresponding to each node k is updated at nodes i and j with the new information available following the exchange (if any).
Let the random variable $D_{ij}(t)$ denote the delay at which the state information of node j is available to node i at time t. The following proposition gives an upper bound on the expectation of the maximum age of any entry across the network:

Proposition 7. *Under the action of the above policy, for any connected network graph $\mathcal{G}$, we have:*

$$\mathbb{E}(\max_{i,j} D_{ij}(t)) \leq \frac{mn}{q}, \;\; \forall t \qquad (18)$$

The above proposition shows that the expected worst case age of state information may be reduced by increasing the parameter q, which, in turn, controls the rate of control information exchange. It is also well-known that the Max-Weight algorithms are robust with respect to delayed queue-length information [20]. Proof of the above proposition is given in Appendix IX-F.

VII. Numerical Simulations

A. Simulating the Throughput-optimal broadcast policy π^*

We simulate the policy π^* on the network $\mathcal{D}_4$, shown in Figure 1. The broadcast-capacity of the network is 2 packets per slot. External packets arrive at the source node $\mathtt{r}$ according to a Poisson process of a slightly lower rate of $\lambda = 1.95$ packets per slot. A packet is said to be broadcast when it reaches all the nodes in the network. The rate of packet arrival and packet broadcast by policy π^*, is shown in Figure 2. This plot exemplifies the throughput-optimality of the policy π^* in the network $\mathcal{D}_4$.

B. Simulating the Multi-class Heuristic Policy π_k^H

The multi-class heuristic policy π_k^H has been numerically simulated in 400 instances of Erdős-Rényi random network

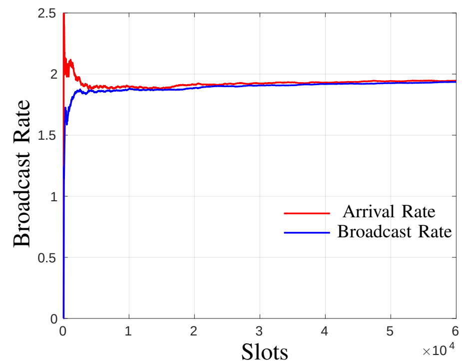

Fig. 2: Packet Arrival and Broadcast Rate in the Diamond Network in Figure 1, under the action of the throughput-optimal policy π^*.

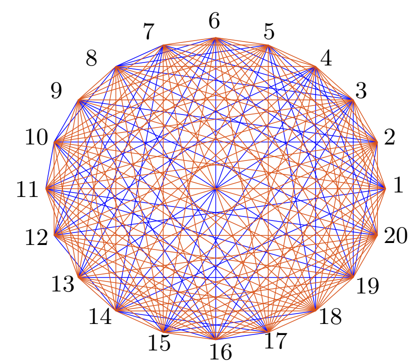

Fig. 3: A network $\mathcal{G}$ with $N = 20$ nodes. The colors of the edges indicate their directions (e.g., *blue edge* $\implies$ $i \to j : i > j$ and vice versa). The broadcast capacity λ^* of the network is computed to be 6, with node 1 being the source node.

with sizes varying from $n = 20$ to $n = 40$ nodes and edge-connectivity probability $p = 0.8$. We have obtained similar qualitative results for all such instances. One representative sample is discussed here.
Consider running the broadcast-policy π_k^H on the network shown in Figure 3, containing $n = 20$ nodes and $m = 176$ edges. The directions of the edges in this network is chosen arbitrarily. With node 1 as the source node, we first compute the broadcast-capacity λ^* of this network using Eqn. (2) and obtain $\lambda^* = 6$. External packets are injected at the source node according to a Poisson process, with a slightly smaller rate of $\lambda = 0.95\lambda^* \approx 5.7$ packets per slot. The rate of broadcast under the multi-class policy π_k^H for different values of k is shown in Figure 4. As evident from the plot, the achievable broadcast rate, obtained by the policy π_k^H, is non-decreasing in the number of classes k. Also, the policy π_k^H broadcasts 95% of the input traffic for a relatively small value of $k = 7$.

C. Minimum Number of Classes for Achieving the Capacity

In this experiment, we simulate the heuristic multiclass policy π_k^H on two different classes of random graphs - Erdős-Rényi and Random Geometric Graphs. We randomly generate 400 instances of Erdős-Rényi graphs from the previous subsection, along with 400 instances of two-dimensional Random Geometric Graphs with $n = 25$ nodes with varying

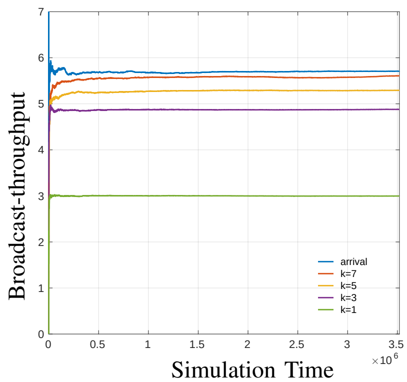

Fig. 4: Achievable broadcast-rate with the multi-class heuristic broadcast-policies π_k^H, for $k = 1, 3, 5, 7$. The underlying network-topology is given in Figure 3 with broadcast capacity $\lambda^* = 6$.

connectivity radii [23]. For each generated graph, we first compute its broadcast capacity λ^* using Theorem 1. Packets arrive at a randomly selected node according to a Poisson process of rate 95% of the computed broadcast capacity of the graph. The empirical average of the minimum number of classes k^* required so that 95% of the incoming packets get broadcasted within $T = 2000$ slots is plotted in Figure 5, along with its coefficient of variation (shown by the little vertical bars). The plot is in excellent agreement with our Conjecture 1, suggesting that for a network with broadcast capacity λ^*, about λ^* classes suffice for achieving near-broadcast-capacity, irrespective of the *size and type* of the network. Figure 5 also suggests that the performance of a fixed number of classes depends on the broadcast capacity of the underlying network.

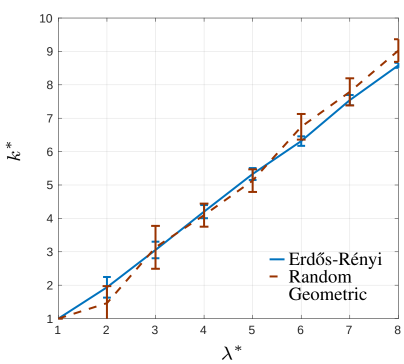

Fig. 5: Number of classes required for achieving 95% of the broadcast capacity in Erdős-Rényi and Random Geometric Graphs.

VIII. Conclusion and Future Directions

In this paper, we studied the problem of efficient, dynamic packet broadcasting in data networks with arbitrary underlying topology. We derived a throughput-optimal Max-weight broadcast policy that achieves the capacity, albeit at the expense of using exponentially many state-variables. To get around this problem, we then proposed a multi-class heuristic policy which combines the idea of in-order packet delivery with a Max-weight scheduling, resulting in drastic reduction in complexity. The proposed heuristic with a small number of classes is conjectured to be throughput-optimal. An immediate next step along this line of work would be to formally prove this conjecture. Another problem of practical interest is to find the minimum number of classes $k^*(\epsilon)$ required to achieve $(1-\epsilon)$ fraction of the capacity.

IX. Appendix

A. Proof of Lemma (1)

Proof: We prove this lemma in two parts. First, we upper-bound the achievable broadcast rate of the network under any policy in the mini-slot model by the broadcast capacity $\lambda^*(\mathcal{G})$ of the network in the usual slotted model, which is given by Eqn. (2). Next, in our main result in section (IX-B), we constructively show that this rate is achievable, thus proving the lemma.

Let $\mathcal{C} \subsetneq V$ be a non-empty subset of the nodes in the graph $\mathcal{G}$ such that $\mathtt{r} \in \mathcal{C}$. Since $\mathcal{C}$ is a strict subset of V, there exists a node $i \in V$ such that $i \in \mathcal{C}^c$. Let the set $E(\mathcal{C})$ denote the set of all directed edges $e = (a, b)$ such that $a \in \mathcal{C}$ and $b \notin \mathcal{C}$. Denote $|E(\mathcal{C})|$ by Cut$(\mathcal{C})$. Using the Max-Flow-Min-Cut theorem [16], the broadcast-capacity in the slotted model, given by Eqn. (2), may be alternatively represented as

$$\lambda^* = \min_{\mathcal{C}\subsetneq V, \mathtt{r}\in\mathcal{C}} \text{Cut}(\mathcal{C}) \tag{19}$$

Now let us proceed with the mini-slot model. Since all packets arrived at source $\mathtt{r}$ that are received by the node i must cross some edge in the cut $E(\mathcal{C})$, it follows that, under any policy $\pi \in \Pi$, the total number of packets $R_i(t)$ that are received by node i up to mini-slot t is upper-bounded by

$$R_i(t) \leq \sum_{\tau=1}^{t} \sum_{e\in E(\mathcal{C})} \mathbb{1}(S(\tau) = e) = \sum_{e\in E(\mathcal{C})} \sum_{\tau=1}^{t} \mathbb{1}(S(\tau) = e) \tag{20}$$

Thus the broadcast-rate $\lambda^\pi_{\text{mini-slot}}$ achievable in the mini-slot model is upper-bounded by

$$\lambda^\pi_{\text{mini-slot}} \overset{(a)}{\leq} \liminf_{t\to\infty} \frac{R_i(t)}{t} \tag{21}$$

$$\overset{(b)}{\leq} \liminf_{t\to\infty} \frac{1}{t} \sum_{e\in E(\mathcal{C})} \sum_{\tau=1}^{t} \mathbb{1}(S(\tau) = e)$$

$$= \sum_{e\in E(\mathcal{C})} \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \mathbb{1}(S(\tau) = e) \tag{22}$$

$$\overset{(c)}{=} \frac{1}{m}\text{Cut}(\mathcal{C}), \text{ w.p.1} \tag{23}$$

Where the inequality (a) follows from the definition of broadcast-rate (1), inequality (b) follows from Eqn. (20) and finally, the equality (c) follows from the Strong Law of Large Numbers [18]. Since the inequality (21) holds for any cut $\mathcal{C} \subsetneq C$ containing the source $\mathtt{r}$ and any policy π, from Eqn. (19) we have

$$\lambda^*_{\text{mini-slot}} \leq \lambda^\pi_{\text{mini-slot}} \leq \frac{1}{m}\text{Cut}(\mathcal{C}) \leq \frac{1}{m}\lambda^* \text{ per mini-slot} \tag{24}$$

Since according to the hypothesis of the lemma, a slot is identified with m mini-slots, the above result shows that

$$\lambda^*_{\text{mini-slot}} \leq \lambda^* \text{ per slot} \tag{25}$$

This proves that the capacity in the mini-slot model (per slot) is at most the capacity of the slotted-time model (given by Eqn. (2)). In section (III), we show that there exists a broadcast policy $\pi^* \in \Pi$ which achieves a broadcast-rate of λ^* packets per-slot in the mini-slot model. This concludes the proof of the lemma. ■

B. Proof of Throughput Optimality of π^*

In this subsection, we show that the induced Markov-Chain $\boldsymbol{Q}^{\pi^*}(t)$, generated by the policy π^* is positive recurrent, for all arrival rates $\lambda < \lambda^*$ packets per slot. This is proved by showing that the expected one-minislot drift of the Lyapunov function $L(\boldsymbol{Q}(t))$ is negative outside a bounded region in the non-negative orthant $\mathbb{Z}_+^M$, where M is the dimension of the state-space $\boldsymbol{Q}(t)$. To establish the required drift-condition, we first construct an auxiliary stationary randomized policy π^{RAND}, which is easier to analyze. Then we bound the one-minislot expected drift of the policy π^* by comparing it with the policy π^{RAND}.

We emphasize that the construction of the randomized policy π^{RAND} is highly non-trivial, because, under the action of the policy π^*, a packet may travel along an arbitrary tree and as a result, *any* reachable set $F \in \mathcal{F}$ may potentially contain a non-zero number of packets.

For ease of exposition, the proof of throughput-optimality of the policy π^* is divided into several parts.

1) Part I: Consequence of Edmonds' Tree-packing Theorem: From Edmond's tree-packing theorem [15], it follows that the graph $\mathcal{G}$ contains λ^* edge-disjoint directed spanning trees, [6] $\{\mathcal{T}^i\}_1^{\lambda^*}$. From Proposition (1) and Lemma (1), it follows that, to prove the throughput-optimality of the policy π^*, it is sufficient to show stochastic-stability of the process $\{\boldsymbol{Q}(t)\}_0^\infty$ for an arrival rate of λ/m per minislot, where $\lambda < \lambda^*$.

Fix an arbitrarily small $\epsilon > 0$ such that,

$$\lambda \leq \lambda^* - \epsilon$$

Now we construct a stationary randomized policy π^{RAND}, which utilizes the edge-disjoint trees $\{\mathcal{T}^i\}_{i=1}^{\lambda^*}$ in a critical fashion.

2) Part II: Construction of a Stationary Randomized Policy π^{RAND} :: The stationary randomized policy π^{RAND} allocates rates $\mu_{e,F}(t)$ randomly to different ordered pairs (e, F), for transmitting packets belonging to reachable sets F, across an edge $e \in \partial^+ F$ [7]. Recall that $\mu_{e,F}(t)$'s are binary variables. Hence, conditioned on the edge-activity process $S(t) = e$, the allocated rates are fully specified by the set of probabilities that a packet from the reachable set F is transmitted across the active edge $e \in \partial^+ F$. Equivalently, we may specify the allocated rates in terms of their expectation w.r.t. the edge-activation process (obtained by multiplying the corresponding probabilities by $1/m$).

Informally, the policy π^{RAND} allocates most of the rates along the reachable sequences corresponding to the edge-disjont spanning trees $\{\mathcal{T}^i\}_1^{\lambda^*}$, obtained in Part I. However, since the dynamic policy π^* is not restricted to route packets along the spanning trees $\{\mathcal{T}^i\}_1^{\lambda^*}$ only, for technical reasons which will be evident later, π^{RAND} is designed to allocate small amount of rates along other reachable sequences. This is an essential and non-trivial part of the proof methodology. An illustrative example of the rate allocation strategy by the policy π^{RAND} will be described subsequently for the diamond graph $\mathcal{D}_4$ of Figure 1.

Formally, the rate-allocation by the randomized policy π^{RAND} is given as follows:

- We index the set of all reachable sequences in a specific order.
  - The first λ^* reachable sequences $\{\zeta^i\}_{i=1}^{\lambda^*}$ are defined as follows: for each edge-disjoint tree $\mathcal{T}^i, i = 1, 2, \ldots, \lambda^*$ obtained from Part-I, recursively construct a reachable sequence $\zeta^i = \{(F^i_j, e^i_j)\}_{j=1}^{n-1}$, such that the induced sub-graphs $\mathcal{T}^i(F^i_j)$ are connected for all $j = 1, 2, \ldots, n-1$.
    In other words, for all $1 \leq i \leq \lambda^*$ define $F^i_1 = \{\mathtt{r}\}$ and for all $1 \leq j \leq n-2$, the set F^i_{j+1} is recursively constructed from the set F^i_j by adding a node to the set F^i_j while traversing along an edge of the tree $\mathcal{T}^i$. Let the corresponding edge in $\mathcal{T}^i$ connecting the $j+1$ th vertex $F^i_{j+1} \setminus F^i_j$, to the set F^i_j, be e^i_j. Since the trees $\{\mathcal{T}^i\}_{i=1}^{\lambda^*}$ are edge disjoint, the edges e^i_j's are **distinct** for all $i = 1, 2, \ldots, \lambda^*$ and $j = 1, 2, \ldots, n-1$. The above construction defines the first λ^* reachable sequences $\zeta^i = \{F^i_j, e^i_j\}_{j=1}^{n-1}, 1 \leq i \leq \lambda^*$.
  - In addition to the above, let $\{\zeta^i = (F^i_j, e^i_j)\}_{j=1}^{n-1}, \lambda^* + 1 \leq i \leq B$ be the set of all *other* reachable sequence in the graph $\mathcal{G}$, different from the previously constructed λ^* reachable sequences. Recall that, B is the cardinality of the set of all reachable sequences in the graph $\mathcal{G}$. Thus the set of *all* reachable sequences in the graph $\mathcal{G}$ is given by $\bigcup_{i=1}^B \zeta^i$.
- To define the expected allocated rates $\mathbb{E}\mu_{e,F}(t)$, it is useful to first define some auxiliary variables, called *rate-components* $\mathbb{E}\mu^i_{e,F}(t), i = 1, 2, \ldots, B$, corresponding to each reachable sequence. The rate $\mathbb{E}\mu_{e,F}(t)$ is is simply the sum of the rate-components, as given in Eqn. (28).
  At each slot t and $1 \leq i \leq \lambda^*$, the randomized policy allocates i^{th} *rate-component* corresponding to the reachable sequence $\zeta^i = \{e^i_j, F^i_j\}_{j=1}^{n-1}$ according to the following scheme:

$$\begin{aligned}\mathbb{E}\big(\mu^i_{e^i_j, F^i_j}(t)\big) &= 1/m - \epsilon(n-j)/n, \\ &\quad \forall\, 1 \leq j \leq n-1 \\ &= 0, \quad \text{o.w.}\end{aligned} \tag{26}$$

[6] Note that, since the edges are assumed to be of unit capacity, λ^* is an integer. This result follows by combining Eqn. (2) with the Max-Flow-Min-Cut theorem [16].

[7] If $e \notin \partial^+ F$, naturally $\mu_{e,F}(t) = 0, \forall t$.

- In addition to the rate-allocation (26), the randomized policy π^{RAND} also allocates small amount of rates corresponding to other reachable sequences $\{\zeta^i\}_{\lambda^*+1}^B$ according to the following scheme: For $\lambda^*+1 \leq i \leq B$, the randomized policy allocates i^{th} rate-component to the ordered pairs (e, F) as follows:

$$\begin{aligned}\mathbb{E}\big(\mu^i_{e^i_j, F^i_j}(t)\big) &= \frac{\epsilon}{2nB} - \frac{\epsilon}{2nB}\frac{n-j}{n}, \\ &\quad \forall\, 1 \leq j \leq n-1, \\ &= 0, \quad \text{o.w.}\end{aligned} \tag{27}$$

The overall rate allocated to the pair (e, F) is simply the sum of the component-rates, as given below:

$$\mathbb{E}\mu_{e,F}(t) = \sum_{i=1}^{B} \mathbb{E}\mu^i_{e,F}(t) \tag{28}$$

In the following, we show that the above rate-allocation is feasible with respect to the edge capacity constraint.

Lemma 2 (**Feasibility of Rate Allocation**). *The rate allocation* (28) *by the randomized policy π^{RAND} is feasible.*

The reader is referred to Appendix (IX-C) for the proof the lemma. An illustrative example for the above randomized rate-allocation scheme is given in Appendix (IX-D).

3) Part III: Comparison of drifts under action of policies π^* and π^{RAND} : Recall that, from Eqn. (13) we have the following upper-bound on the one-minislot drift of the Lyapunov function $L(\boldsymbol{Q}(t)$, achieved by the policy π^*:

$$(\Delta^{\pi^*}(\boldsymbol{Q}(t)|S(t)) \leq 2^n\mu^2_{\max} - \sum_{(e,F):e\in\partial^+F}\Big(Q_F(t) - Q_{F+e}(t)\Big)\mathbb{E}\big(\mu^{\pi^*}_{e,F}(t)|\boldsymbol{Q}(t), S(t)\big)$$

Since the policy π^*, by definition, transmits packets to maximize the weight $w_{F,e}(t) = Q_F(t) - Q_{F+e}(t)$ point wise, the following inequality holds

$$\sum_{(e,F):e\in\partial^+F}\Big(Q_F(t) - Q_{F+e}(t)\Big)\mathbb{E}\big(\mu^{\pi^*}_{e,F}(t)|\boldsymbol{Q}(t), S(t)\big) \geq \sum_{(e,F):e\in\partial^+F}\Big(Q_F(t) - Q_{F+e}(t)\Big)\mathbb{E}\big(\mu^{\pi^{\text{RAND}}}_{e,F}(t)|\boldsymbol{Q}(t), S(t)\big),$$

where the randomized rate-allocation $\boldsymbol{\mu}^{\pi^{\text{RAND}}}$ is given by Eqn. (28). Noting that π^{RAND} operates independently of the "queue-states" $\boldsymbol{Q}(t)$ and dropping the super-script π^{RAND} from the control variables $\boldsymbol{\mu}(t)$ on the right hand side, we can bound the one-slot expected drift of the policy π^* as follows:

$$\begin{aligned}&(\Delta^{\pi^*}(\boldsymbol{Q}(t))|S(t)) \\ \leq\;& 2^n\mu^2_{\max} - \sum_{(e,F):e\in\partial^+F}\Big(Q_F(t) - Q_{F+e}(t)\Big)\mathbb{E}\big(\mu_{e,F}(t)|S(t)\big) \\ =\;& 2^n\mu^2_{\max} - \sum_F Q_F(t)\Big(\sum_{e\in\partial^+F}\mathbb{E}(\mu_{e,F}(t)|S(t)) \\ &- \sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\mathbb{E}(\mu_{e,G}(t)|S(t))\Big) \\ \stackrel{(a)}{=}\;& 2^n\mu^2_{\max} - \sum_F Q_F(t)\Big(\sum_{e\in\partial^+F}\big(\sum_{i=1}^B\mathbb{E}(\mu^i_{e,F}(t)|S(t))\big) - \\ &\sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\big(\sum_{i=1}^B\mathbb{E}(\mu^i_{e,G}(t)|S(t))\big)\Big),\end{aligned}$$

where in (a) we have used Eqn. (28).
Taking expectation of both sides of the above inequality w.r.t the random edge-activation process $S(t)$ and interchanging the order of summation, we have

$$\begin{aligned}\Delta^{\pi^*}(\boldsymbol{Q}(t)) \leq\;& 2^n\mu^2_{\max} - \sum_F Q_F(t)\sum_{i=1}^B\Big(\sum_{e\in\partial^+F}\mathbb{E}(\mu^i_{e,F}(t)) \\ &- \sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\mathbb{E}(\mu^i_{e,G}(t))\Big),\end{aligned} \tag{29}$$

where the rate-components $\boldsymbol{\mu}^i$ of the randomized policy π^{RAND} are defined in Eqns (26) and (27).
Fix a reachable set F, appearing in the outer-most summation of the above upper-bound (29). Now focus on the i^{th} reachable sequence $\zeta^i \equiv \{F^i_j, e^i_j\}_1^{n-1}$. We have two cases:
Case I: $F \notin \zeta^i$
Here, according to the allocations in (26) and (27), we have

$$\sum_{e\in\partial^+F}\mathbb{E}(\mu^i_{e,F}(t)) \stackrel{(a)}{=} 0, \qquad \sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\mathbb{E}(\mu^i_{e,G}(t)) \stackrel{(b)}{=} 0$$

Where the equality (a) follows from the assumption that $F \notin \zeta^i$ and equality (b) follows from the fact that positive rates are allocated only along the tree corresponding to the reachable sequence ζ^i. Hence, if no rate is allocated to drain packets outside the set F, π^{RAND} does not allocate any rate to route packets to the set F.

Case II: $F \in \zeta^i$
In this case, from Eqns. (26) and (27), it follows that

$$\Big(\sum_{e\in\partial^+F}\mathbb{E}(\mu^i_{e,F}(t)) - \sum_{(e,G):e\in\partial^-F, G=F\setminus\{e\}}\mathbb{E}(\mu^i_{e,G}(t))\Big) = \begin{cases}\frac{\epsilon}{n}, & 1 \leq i \leq \lambda^* \\ \frac{\epsilon}{2n^2B}, & \lambda^*+1 \leq i \leq B\end{cases} \tag{30}$$

By definition, each reachable set is visited by at least one reachable sequence. In other words, there exists at least one $i, 1 \leq i \leq B$, such that $F \in \zeta^i$. Combining the above two cases, from the upper-bound (29) we conclude that

$$\Delta^{\pi^*}(\boldsymbol{Q}(t)) \leq 2^n\mu^2_{\max} - \frac{\epsilon}{2n^2B}\sum_F Q_F(t), \tag{31}$$

where, the sum extends over *all* reachable sets. The drift is negative, i.e., $\Delta^{\pi^*}(\boldsymbol{Q}(t)) < -\epsilon$, when $\boldsymbol{Q}_F \in \mathcal{B}^c$, where

$$\mathcal{B} = \Big\{(Q_F \geq 0) : \sum_F Q_F \geq \frac{2n^2B}{\epsilon}(\epsilon + 2^n\mu^2_{\max})\Big\}$$

Invoking the Foster-Lyapunov criterion [21], we conclude that the Markov-Chain $\{\boldsymbol{Q}^{\pi^*}(t)\}_0^\infty$ is positive recurrent. Finally, throughput-optimality of the policy π^* follows from lemma 1. ■

C. *Proof of Lemma* (2)

Proof: The rate allocation (28) will be feasible if the sum of the allocated probabilities that an active edge e carries a class-F packet, for all reachable sets F, is at most unity. Since an edge can carry at most one packet per mini-slot, this feasibility condition is equivalent to the requirement that the total expected rate, i.e., $\mathbb{E}\mu_e(t) = \sum_F \mathbb{E}\mu_{e,F}(t)$, allocated to an edge $e \in E$ by the randomized policy π^{RAND} does not exceed $\frac{1}{m}$ (the expected capacity of the edge per mini-slot). Since an edge e may appear at most once in any reachable sequence, the total rate allocated to an edge e by the randomized-policy π^{RAND} is upper-bounded by $\frac{1}{m} - \frac{\epsilon}{n} + (B-\lambda^*)\frac{\epsilon}{2nB} \leq \frac{1}{m} - \frac{\epsilon}{2n} < 1/m$. Hence the rate allocation by the randomized policy π^{RAND} is feasible. ■

D. *An Example of Rate Allocation by the Stationary policy π^{RAND}*

As an explicit example of the above stationary policy, consider the case of the Diamond network $\mathcal{D}_4$, shown in Figure 1. The edges of the trees $\{\mathcal{T}^i, i=1,2\}$ are shown in blue and red colors in the figure. Then the randomized policy allocates the following rate-components to the edges, where the expectation is taken w.r.t. random edge-activations per mini-slot.

First we construct a reachable sequence ζ^1 consistent with the tree $\mathcal{T}^1$ as follows:

$$\zeta^1 = \{(\{\texttt{r}\}, \texttt{ra}), (\{\texttt{r,a}\}, \texttt{ab}), (\{\texttt{r,a,b}\}, \texttt{bc})\}$$

Next we allocate the following rate-components as prescribed by π^{RAND}:

$$\begin{aligned}
\mathbb{E}\mu^1_{\texttt{ra},\{\texttt{r}\}}(t) &= 1/6 - 3\epsilon/4\\
\mathbb{E}\mu^1_{\texttt{ab},\{\texttt{r,a}\}}(t) &= 1/6 - 2\epsilon/4\\
\mathbb{E}\mu^1_{\texttt{bc},\{\texttt{r,a,b}\}}(t) &= 1/6 - \epsilon/4\\
\mathbb{E}\mu^1_{e,F}(t) &= 0, \quad \text{o.w.}
\end{aligned}$$

Similarly for the tree $\mathcal{T}^2$, we first construct a reachable sequence ζ^2 as follows:

$$\zeta^2 = \{(\{\texttt{r}\}, \texttt{rb}), (\{\texttt{r,b}\}, \texttt{rc}), (\{\texttt{r,b,c}\}, \texttt{ca})\}$$

Then we allocate the following component-rates to the (edge, set) pairs as follows:

$$\begin{aligned}
\mathbb{E}\mu^2_{\texttt{rb},\{\texttt{r}\}}(t) &= 1/6 - 3\epsilon/4\\
\mathbb{E}\mu^2_{\texttt{rc},\{\texttt{r,b}\}}(t) &= 1/6 - 2\epsilon/4\\
\mathbb{E}\mu^2_{\texttt{ca},\{\texttt{r,b,c}\}}(t) &= 1/6 - \epsilon/4\\
\mathbb{E}\mu^2_{e,F}(t) &= 0, \quad \text{o.w.}
\end{aligned}$$

In this example $\lambda^* = 2$, thus these two reachable sequence accounts for a major portion of the rates allocated to the edges. The randomized policy π^{RAND}, however, allocates small rates to other reachable sequences too. As an example, consider the following reachable sequence ζ^3, given by

$$\zeta^3 = \{(\{\texttt{r}\}, \texttt{ra}), (\{\texttt{r,a}\}, \texttt{rb}), (\{\texttt{r,a,b}\}, \texttt{rc}\}$$

Then, as prescribed above, the randomized policy allocates the following rate-components

$$\begin{aligned}
\mathbb{E}\mu^3_{\texttt{ra},\{\texttt{r}\}}(t) &= \frac{\epsilon}{8B} - \frac{3\epsilon}{32B}\\
\mathbb{E}\mu^3_{\texttt{rb},\{\texttt{r,a}\}}(t) &= \frac{\epsilon}{8B} - \frac{2\epsilon}{32B}\\
\mathbb{E}\mu^3_{\texttt{rc},\{\texttt{r,a,b}\}}(t) &= \frac{\epsilon}{8B} - \frac{\epsilon}{32B}\\
\mathbb{E}\mu^3_{e,F}(t) &= 0, \quad \text{o.w.}
\end{aligned}$$

Here B is the number of all distinct reachable sequences, which is upper-bounded by 4^8. The rate-components corresponding to other reachable sequences may be computed as above. Finally, the actual expected rate-allocation to the pair (e,F) is given by

$$\mathbb{E}\mu_{e,F}(t) = \sum_{i=1}^{B} \mathbb{E}\mu^i_{e,F}(t)$$

E. *Proof of Proposition* (4)

The proof of this proposition is conceptually simplest in the slotted-time model. The argument also applies directly to the mini-slot model.

Consider a network $\mathcal{G}$ with broadcast-capacity λ^*. Assume a slotted-time model. By Edmonds' tree-packing Theorem [15], we know that there exists λ^* number of edge-disjoint directed spanning trees (arborescences) $\{\mathcal{T}_i\}_1^{\lambda^*}$ in $\mathcal{G}$, rooted at the source node $\texttt{r}$. Now consider a policy $\pi \in \Pi_k^{\text{in-order}}$ with $k \geq \lambda^*$ which operates as follows:

- An incoming packet is placed in any of the classes $[1, 2, 3, \ldots, \lambda^*]$, uniformly at random.
- Packets in a class i are routed to all nodes in the network *in-order* along the directed tree $\mathcal{T}_i$, where the packets are replicated in all non-leaf nodes of the tree $\mathcal{T}_i, 1 \leq i \leq \lambda^*$.

Since the trees are edge-disjoint, the classes do not conflict; i.e., routing in each class can be carried out independently. Also by the property of $\mathcal{T}_i$, there is a *unique* directed path from the source node $\texttt{r}$ to any other node in the network along the edges of the tree $\mathcal{T}_i, 1 \leq i \leq \lambda^*$. Thus packets in every class can be delivered to all nodes in the network *in-order* in a pipe-lined fashion with the long-term delivery-rate of 1 packet per class. Since there are λ^* packet-carrying classes, it follows that the policy $\pi \in \Pi_k^{\text{in-order}}$ is throughput-optimal for $k \geq \lambda^*$.

Next we show that, $\lambda^* \leq n/2$ for a simple network. Since there exist λ^* number of edge-disjoint directed spanning trees in the network, and since each spanning-tree contains $n-1$ edges, we have

$$\lambda^*(n-1) \leq m \tag{32}$$

Where m is the number of edges in the network. But we have $m \leq n(n-1)/2$ for a simple graph. Thus, from the above equation, we conclude that

$$\lambda^* \leq n/2. \tag{33}$$

This completes the proof of the Proposition.

F. *Proof of Proposition* (7)

Proof: Consider a spanning tree $\mathcal{T}$ in the network (it exists, as the network is assumed to be connected). One of the many possible ways to send its state information from any node j to a node i would be to send this information following the unique path $\mathcal{P}_{ij}$ induced by the tree $\mathcal{T}$. Thus,

$$D_{ij}(t) \leq \sum_{e \in \mathcal{P}_{ij}} X_e, \tag{34}$$

where X_e is the (stationary) random variable denoting the number of required minislots until a state exchange takes place along the edge e. Hence, it follows that

$$\max_{i,j} D_{ij}(t) \leq \max_{i,j} \sum_{e \in \mathcal{P}_{ij}} X_e \overset{(a)}{\leq} \sum_{e \in \mathcal{T}} X_e, \tag{35}$$

where the inequality (a) follows from the fact that the r.v.s X_e's are non-negative. Since X_e's are geometrically distributed with parameter $\frac{q}{m}$, we have $\mathbb{E}X_e = \frac{m}{q}$. Taking expectation of both sides of (35), we have

$$\mathbb{E}\big(\max_{i,j} D_{ij}(t)\big) \overset{(a)}{\leq} \sum_{e \in \mathcal{T}} \mathbb{E}X_e \leq \frac{mn}{q},$$

where the inequality (a) follows from the fact that there are exactly $n-1$ edges in the spanning tree $\mathcal{T}$. ■

References

[1] A. Sinha, G. Paschos, and E. Modiano, "Throughput-optimal multi-hop broadcast algorithms," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16. New York, NY, USA: ACM, 2016, pp. 51–60. [Online]. Available: http://doi.acm.org/10.1145/2942358.2942390

[2] J. P. Macker, J. E. Klinker, and M. S. Corson, "Reliable multicast data delivery for military networking," in *Military Communications Conference, 1996. MILCOM'96, Conference Proceedings, IEEE*, vol. 2. IEEE, 1996, pp. 399–403.

[3] R. Chen, W.-L. Jin, and A. Regan, "Broadcasting safety information in vehicular networks: issues and approaches," *IEEE network*, vol. 24, no. 1, pp. 20–25, 2010.

[4] X. Zhang, J. Liu, B. Li, and Y.-S. Yum, "Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 3. IEEE, 2005, pp. 2102–2111.

[5] S. Jiang, L. Guo, and X. Zhang, "Lightflood: an efficient flooding scheme for file search in unstructured peer-to-peer systems," in *Parallel Processing, 2003. Proceedings. 2003 International Conference on*, Oct 2003, pp. 627–635.

[6] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1110–1122, 1996.

[7] A. Bar-Noy, S. Kipnis, and B. Schieber, "Optimal multiple message broadcasting in telephone-like communication systems," *Discrete Applied Mathematics*, vol. 100, no. 1–2, pp. 1–15, 3 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166218X99001559

[8] 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.

[9] J. L. Träff and A. Ripke, "Optimal broadcast for fully connected networks," *High Performance Computing and Communications: First International Conference, HPCC 2005, Sorrento, Italy, September 21-23, 2005. Proceedings*, 2005.

[10] A. Sinha, G. Paschos, C. P. Li, and E. Modiano, "Throughput-optimal multihop broadcast on directed acyclic wireless networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 377–391, Feb 2017.

[11] A. Sinha, L. Tassiulas, and E. Modiano, "Throughput-optimal broadcast in wireless networks with dynamic topology," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16. New York, NY, USA: ACM, 2016, pp. 21–30. [Online]. Available: http://doi.acm.org/10.1145/2942358.2942389

[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] 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.

[14] D. Towsley and A. Twigg, "Rate-optimal decentralized broadcasting: the wireless case," Citeseer, 2008.

[15] R. Rustin, *Combinatorial Algorithms*. Algorithmics Press, 1973.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[17] 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.

[18] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2010.

[19] D. V. Lindley, "The theory of queues with a single server," in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48. Cambridge Univ Press, 1952, pp. 277–289.

[20] 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.

[21] E. Wong and B. Hajek, *Stochastic processes in engineering systems*. Springer Science & Business Media, 2012.

[22] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.

[23] M. Penrose, *Random geometric graphs*. Oxford University Press, 2003, no. 5.

Abhishek Sinha Abhishek Sinha is currently a doctoral candidate in the Laboratory for Information and Decision Systems, at the Massachusetts Institute of Technology. Prior to joining MIT, he received his M.E. degree (in 2012) from the Indian Institute of Science, Bangalore, and B.E. degree from Jadavpur University, Kolkata (in 2010), all in the field of Electronics and Telecommunication Engineering. He is a recipient of several academic awards including the Best Paper Award in the conference ACM MobiHoc 2016, Jagadis Bose National Science Talent Search scholarship (2007), and Prof. Jnansaran Chatterjee Memorial Gold Medal from Jadavpur University (2010). His research interests include Optimization, Information Theory, and Network Control.

Georgios Paschos Since Nov 2014, Georgios is a principal researcher at Huawei Technologies, Paris, France, leading the Network Control and Resource Allocation team. Previously, he spent two years at MIT in the team of Prof. Eytan Modiano. For the period June 2008-Nov 2014 he was affiliated with "The Center of Research and Technology Hellas - Informatics & Telematics Institute" CERTH- ITI, Greece, working with Prof. Leandros Tassiulas. He also taught in the University of Thessaly, Dept. of Electrical and Computer Engineering as an adjunct lecturer for the period 2009-2011. In 2007-2008 he was an ERCIM Postdoc Fellow in VTT, Finland, working on the team of Prof. Norros. He received his diploma in Electrical and Computer Engineering (2002) from Aristotle University of Thessaloniki, and his PhD degree in Wireless Networks (2006) from ECE dept. University of Patras (supervisor Prof. Stavros Kotsopoulos), both in Greece. Two of his papers won the best paper award, in GLOBECOM 07 and IFIP Wireless Days 09 respectively. He serves as an associate editor for IEEE/ACM Trans. on Networking, and as a TPC member of IEEE INFOCOM.

Eytan Modiano Eytan Modiano received his B.S. degree in Electrical Engineering and Computer Science from the University of Connecticut at Storrs in 1986 and his M.S. and PhD degrees, both in Electrical Engineering, from the University of Maryland, College Park, MD, in 1989 and 1992 respectively. He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Post Doctoral Fellow during 1992-1993. Between 1993 and 1999 he was with MIT Lincoln Laboratory. Since 1999 he has been on the faculty at MIT, where he is a Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks. He is the co-recipient of the Sigmetrics 2006 Best paper award and the WiOpt 2013 best paper award. He is an Editor-at-Large for IEEE/ACM Transactions on Networking, and served as Associate Editor for IEEE Transactions on Information Theory and IEEE/ACM Transactions on Networking. He was the Technical Program co-chair for IEEE Wiopt 2006, 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.