Broadcasting Real-Time Flows in Integrated Backhaul and Access 5G Networks

Aria HasanzadeZonuzy*, I-Hong Hou* and Srinivas Shakkottai*
*Dept. of Electrical and Computer Engineering, Texas A&M University
Email:{azonuzy, ihou, sshakkot}@tamu.edu

Abstract—We study the problem of broadcasting real-time flows in multi-hop wireless networks. We assume that each packet has a stringent deadline, and each node in the network obtains some utility based on the number of packets delivered to it on time for each flow. We propose a distributed protocol called the delegated-set routing (DSR) that incurs virtually no overhead of coordination among nodes. We also develop distributed algorithms that aim to maximize the total timely utility under DSR. The utility of the DSR protocol and distributed algorithms are demonstrated by both theoretical analysis and simulation results. We show that our algorithms achieve higher timely throughput even when compared against centralized throughput optimal policies that do not consider deadline constraints.

I. INTRODUCTION

Mutli-hop broadcasting in wireless networks, which entails disseminating information to every device in the system via retransmissions at multiple nodes, is an important mechanism to coordinate devices in networked systems. Furthermore, many applications of broadcast communications are safety-critical, and timely deliveries of information is crucial to maintain the robustness and safety of the system. For example, multi-hop broadcasting is needed to disseminate timely safety information among connected vehicles in vehicular ad hoc networks (VANETs), to announce control decisions in networked control systems and Internet of Things (IoT), and to exchange locations and flight paths among unmanned aerial vehicles (UAVs) for Unmanned Aircraft System Traffic Management (UTM).

The cellular infrastructure that will enable these time-critical broadcast wireless applications will be 5G networks that are currently being designed to support ultra-low latency, ultra-high throughput communications. These networks will utilize the highly directional and high bandwidth mm-wave band, which suffers from high attenuation and sensitively to fading. This requires the relatively dense deployment of small base stations at spacings of about 250 m. However, providing fiber backhaul to all of these base stations is prohibitively expensive. An important development in this context is *Integrated Access and Backhaul* (IAB) [1], [2], under which there

are a few base stations with fiber backhaul that act as gateways to many others that are connected via a mm-wave wireless mesh backhaul. This mm-wave backhaul creates a directional wireless network between the nodes, but routing across these is highly dynamic and subject to the vagaries of the wireless channel. The same mm-wave spectrum also is used to provide access to end-users, i.e., both access and backhaul are integrated over mm-wave.

Motivated by the above features of emerging networks, this paper studies the problem of designing algorithms for broadcasting real-time flows with strict per-packet end-to-end deadlines in directional wireless mesh networks. Here, real-time flow imposes a strict deadline for each of its packets, and packets that cannot be delivered before their respective deadlines are dropped from the system. From the IAB perspective, our goal is to ensure that each broadcast packet is delivered to an appropriate IAB base station before its deadline, at which point it is immediately transmitted to its respective end user. Each IAB node in the network then obtains some utility based on the time-average number of on-time packets that it receive from each flow. The goal of this paper is to maximize the total timely-utility of the whole network.

There are several important challenges that need to be addressed for broadcasting real-time flows in such multi-hop mmWave networks. First, since it is difficult to coordinate a large network in real-time, centralized algorithms that require the instant knowledge of the state of each node and packet are usually infeasible to implement. Hence, we need distributed algorithms, where each node makes decisions using its local information. Second, as mentioned above, transmissions in the mmWave band can be unreliable. Finally, broadcasting algorithms need to explicitly address the deadline requirement of each flow.

Main Results and Organization

In this paper, we propose a new protocol for broadcasting in multi-hop mmWave networks, namely, the *delegated-set routing* (DSR) protocol. DSR has two important features: First, it is a distributed protocol where all the required coordination among nodes can be conveyed in the headers of packets once the topology of the network is known. Hence, there is virtually no overhead of coordination after topology creation process. Second, DSR

allows each node to dynamically change its transmission strategies based on the deadlines of its packets and random events, such as transmission failures, it experiences.

Relaxing the link utilization constraint (number of tranmissions allowed per time slot) to an average one, and using dual decomposition techniques, we also propose a distributed algorithm that aims to maximize the total system-wide utility under DSR. This algorithm only requires minimal and infrequent information exchange among nodes. We analytically prove that our algorithm achieves the optimal total utility under an average link capacity constraint. The key novelty lies in a natural decomposition into packet-by-packet and link-by-link updates that need minimal coordination. These lead to a steepest-ascent-type control associated with each packet, and a sub-gradient type of update at links. This algorithm also gives rise to a simple index policy when link utilization constraints of all links need to be satisfied at every instant.

We evaluate our algorithms through simulations on representative network graphs. We compare our algorithms against recent studies on throughput optimal algorithms, including one that is designed specifically for broadcast, and one that is universal in terms of being able to support unicast, multicast and broadcast. We show that despite some of these algorithms being centralized and complex, our algorithm, which is designed specifically for simplicity and delay optimality, achieves better performance.

The paper is organized as follows. Section II reviews existing studies on broadcasting and multi-hop networks. Section III describes our system model for multi-hop networks with real-time broadcast flows. Section IV describes the additional structure imposed by the DSR protocol, as well as an epoch-wise approach to policy selection. Section V applies dual-decomposition, which turns out to be the basis of our distributed algorithm. Section VI proposes distributed algorithms that optimize DSR, as well as the index policy that can ensure hard capacity constraints are met. Section VII presents our simulation results. Finally, Section VIII concludes the paper.

II. RELATED WORK

Broadcasting/multicasting is a fundamental functionality of networks, and has been studied in a substantial body of literature. One of the earliest policies for broadcasting/multicasting in ad hoc networks is via flooding [3], [4]. However, such policies can lead to severe packet collision frequency, and excessive redundant retransmissions, as shown by Ni et al. [5]. Gandhi et al. [6] and Huang et al. [7] have shown that the problem of minimizing delay in wireless ad hoc networks is NP-hard, and have proposed approximation algorithms aiming to reduce delay. These studies rely on centralized algorithms.

There has been much interest in throughput optimal broadcasting/multicasting. For instance, Sarkar and Tassiulas [8] proposed a scheduling and routing policy that relies on pre-computed spanning trees, which might be diffi-cult to maintain and compute in scalable sized networks. Ho and Viswanathan [9] and Yuan et al. [10] propose network coding based policies in the context, which, however, leads to additional computation complexity. Zhang et al. [11] and Sinha et al. [12] consider multi-hop broadcasting problems in Directed Acyclic Graphs (DAG), which are not applicable to networks with arbitrary topology. Sinha et al. [13] also propose a centralized throughput optimal broadcasting policy for networks with arbitrary topology, which might be difficult to deploy in a large scale system. Furthermore, the throughput maximization focus of all the above does not directly allow for meeting stringent deadline guarantees.

Given the rising application of wireless networks to safety-critical and realtime applications, there has been much recent interest in deadline constrained multi-hop communication. Xiong et al. [14] proposed a delay-aware throughput optimal policy for multi-hop networks. Their policy, however, can not provide stringent delay guarantees. Mao et al. [15] propose a hard deadline guaranteed policy, under the assumption that all routes in the network are fixed. Li and Eryilmaz [16] consider serving flows with stringent deadlines in a multi-hop system, and their proposed framework can be extended to incorporate routing decisions. However, their policies are heuristic, and optimality cannot be shown. Singh and Kumar [17] relax the deadline constrained optimization problem in the manner of the Whittle's relaxation for multi-armed bandits, and proposed decentralized optimal solutions. However, both it and the above body of work on deadline constrained communication only considers unicast traffic, and it is not clear how it applies to broadcasting/multicasting networks.

III. SYSTEM MODEL

We consider a multi-hop network that consists of N wireless nodes operating in the mmWave band motivated by the IAB system. Here, the nodes correspond to fixed IAB base stations, and the network topology is known to all nodes. The available spectrum is divided into multiple half-duplex channels, and nodes can use these channels to send and receive packets from multiple nodes simultaneously. Furthermore, these channels are *directional* in that transmissions on different links do not interfere with each other, consistent with empirical observations in IAB test deployments [2]. These links can have different constraints on the supportable number of transmissions in each time slot, as well as their reliabilities.

Time is slotted and numbered as $t = 1, 2, \ldots$. We assume that link l can transmit T_l packets in each time slot, and that each transmission will be successfully received by the receiver with probability P_l. At the end of each time slot, the receiver sends an aggregated ACK indicating which packets it has successfully received in the time slot to the transmitter. Where we need to indicate the

transmitter and the receiver of a link, we use $l = n \to m$ to indicate that link l has transmitter n and receiver m.

We consider F real-time broadcast flows, using s_f to indicate the source node of flow f. At the beginning of each time slot t, $a_f(t)$ packets of flow f arrive at node s_f. We assume that $[a_f(1), a_f(2), \dots]$ is a sequence of i.i.d. random variables with mean A_f. Moreover, each flow f specifies a per-packet end-to-end deadline of D_f time slots. Packets from flow f are only useful for D_f time slots from their respective arrival times at their source nodes, and are dropped from the network when they expire. Due to communication constraints, it is likely that some nodes cannot receive all packets from each flow. We therefore measure the performance of node n on flow f by its *timely-throughput*, defined as the long-term average number of packets from flow f that are successfully delivered to node n within the deadline per time unit.

Let Ω be a set of stationary packet scheduling policies. Hence, given the state of the system consisting of the locations and expiry times of all existing packets, a policy $\omega \in \Omega$ is a rule that decides which packet to transmit on what link, subject to communication constraints. For each stationary policy $\omega \in \Omega$, let $x^\omega_{n,f}(t)$ be the number of packets from f that are delivered to n at time t under ω, i.e., these are the packets that survived the deadline constraint. Also, let $\epsilon^\omega_{l,f}(t)$ be the number of packets from flow f transmitted over link l at time t under ω. Since ω is a stationary policy, and all packets that expire are immediately dropped, we can define

$$\mu^\omega_{n,f} := \liminf_{T \to \infty} \frac{\sum_{t=1}^{T} x^\omega_{n,f}(t)}{T}$$

as the timely-throughput of node n on flow f under ω, and

$$\bar{\epsilon}^\omega_{l,f} := \limsup_{T \to \infty} \frac{\sum_{t=1}^{T} \epsilon^\omega_{l,f}(t)}{T}$$

as the average number of transmissions for flow f over link l under ω.

Now, finding the optimal total utility with respect to timely-throughputs over all the N nodes under DSR is equivalent to finding the stationary policy that maximizes the total timely-utility under link utilization constraints, which can be written as

Relaxed Timely-Utility Maximization (R-TUM)

$$\text{Max } \sum_{n=1}^{N} \sum_{f=1}^{F} U_{n,f}(\mu^\omega_{n,f}) \tag{1}$$

$$\text{s.t. } \omega \in \Omega, \tag{2}$$

$$\sum_{f=1}^{F} \bar{\epsilon}^\omega_{l,f} \leq T_l, \forall l. \tag{3}$$

Notice that whereas the R-TUM problem above requires each delivered packet to satisfy its deadline constraint, it

only requires that the *long-term average* number of transmissions over link l, $\sum_{f=1}^{F} \bar{\epsilon}^\omega_{l,f}$ be no larger than T_l. This link utilization constraint relaxation is in the same manner as [17]. In a practical system, such a relaxation might be akin to imposing an average transmit power constraint rather than a hard one. We will first design policies that pertain to this relaxed link-utilization constraint. Using the insights gained, we will also develop a policy that enforces a *hard link-utilization constraint,* i.e.,

$$\sum_{f=1}^{F} \epsilon^\omega_{l,f}(t) \leq T_l, \forall l, t. \tag{4}$$

Solving the R-TUM problem could be posed as a Markov Decision Process (MDP), where the state of the system at any given point of time consists of the locations and expiry times of all existing packets. However, such a solution is infeasible to implement in practice. First, it is straightforward to show that the number of different system states is at least doubly exponential in N, and hence standard algorithms for finding the optimal MDP-based solution will result in prohibitive complexity. Second, even after one finds the optimal MDP-based solution, it may be impossible to implement it in a distributed fashion, since the complete state needs to be known at each node. In what follows, we impose additional structure on the policy space to render it tractable.

IV. A STRUCTURED APPROACH TO REAL-TIME BROADCASTING

We now introduce two elements of structure to the policy space to enable its solution as a distributed convex optimization problem.

A. Delegated-Set Routing (DSR)

Ensuring a per-packet deadline guarantee requires that we retain flexibility in routing to dynamically choose the next hop node for a packet based on current state. Thus, source routing on a per-packet basis is not satisfactory. However, for distributed implementation, we also need to ensure that there is no ambiguity as to which neighboring node is responsible for transmitting a packet to a given node. We resolve these seemingly opposite requirements via a protocol that we term *delegated-set routing (DSR)*.

For each node n that possesses a packet i at time t, we define the delegated-set of node n as the subset of nodes that n is responsible for forwarding packets, possibly through multi-hop transmissions. First, to ensure routing flexibility, whenever a node n decides to forward a packet to another node m, node n delegates a subset of its own delegated-set to m, and specifies this subset in the packet header. If the transmission is successful, this subset is removed from the delegated-set of n, since it is now the responsibility of m to forward the packet to this subset. Second, in order to avoid duplicate transmissions (ambiguity on which node should transmit a given

packet), the DSR protocol requires that the delegated-sets of different nodes for the same packet are chosen to be disjoint.
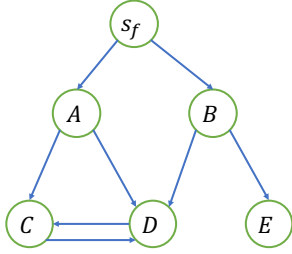

Fig. 1. An example illustrating DSR.

To illustrate how DSR works, consider the network as shown in Fig. 1. When a packet arrives at the source node s_f, the delegated set of s_f is every node in the network, since it is the responsibility of s_f to broadcast the packet to the entire network. Suppose in the first time slot, s_f transmits the packet to A, and delegates the subset $\{A, C, D\}$ to A. If the transmission is successful, the delegated-set of s_f becomes $\{s_f, B, E\}$, while the delegated-set of A is $\{A, C, D\}$. In the next time slot, when s_f transmits the packet to B, it needs to delegate the subset $\{B, E\}$ to B. In particular, s_f cannot include node D in the delegated-set for B, since D is already in the delegated-set of A.

We note that the ability to dynamically adjust routing decisions is an important feature that distinguishes DSR from many existing studies on multi-hop broadcasting, such as [13] and [8]. These studies adopt source-routing, where the source node determines the routing decision of each packet, and intermediate nodes cannot change the decision. As s_f cannot foresee whether the transmissions from A to C will be successful, it cannot take an optimal routing decision.

B. Epoch-wise Stationary Policies

Our second aspect of adding structure to the policy space is to expand it from Ω, the set of all stationary policies, to the set of all *epoch-wise stationary policies*. In an epoch-wise stationary policy, time is divided into epochs of equal length. The epoch-wise stationary policy adopts a stationary policy $\omega^+[i]$ in each epoch i. The duration of an epoch is chosen to be large enough so that the average performance of $\omega^+[i]$ in epoch i is not influenced by the system state at the beginning of the epoch. Specifically, an epoch-wise stationary policy is defined as follows:

Definition 1. *An epoch-wise stationary policy is a sequence of stationary policies $\omega^+ = (\omega[i])_{i=1}^{\infty}, \omega[i] \in \Omega)$, where $\omega[i]$ is used in epoch i. The length of an epoch is chosen so that, under ω^+,*

$$\mu_{n,f}^{\omega^+} := \liminf_{T \to \infty} \frac{\sum_{t=1}^{T} x_{n,f}^{\omega^+}(t)}{T} = \liminf_{I \to \infty} \frac{\sum_i^I \mu_{n,f}^{\omega[i]}}{I},$$

and

$$\bar{\epsilon}_{l,f}^{\omega^+} := \limsup_{T \to \infty} \frac{\sum_{t=1}^{T} \epsilon_{l,f}^{\omega^+}(t)}{T} = \limsup_{I \to \infty} \frac{\sum_i^I \bar{\epsilon}_{l,f}^{\omega[i]}}{I}.$$

We can now define Ω^+ as the set of all epoch-wise stationary policies. For each epoch-wise stationary policy ω^+, let $\gamma^{\omega^+} := [[\mu_{n,f}^{\omega^+}, 1 \le n \le N, 1 \le f \le F], [\bar{\epsilon}_{l,f}^{\omega^+}, 1 \le l \le L, 1 \le f \le F]]$ be the vector of timely-throughputs and average link uses under ω^+. Also, let $\Gamma := \{\gamma^{\omega^+} | \omega^+ \in \Omega^+\}$ be the set of attainable vectors of timely-throughputs and average link uses under all epoch-wise stationary policies. An important advantage of considering the policy space Ω^+ is that Γ is a convex set.

Lemma 1. Γ *is convex.*

Proof. The proof is is provided in [18]. $\square$

Since Γ is a convex set, it is straightforward to verify that the optimization problem (1)–(3) subject the policy space Γ is a convex optimization problem.

V. SOLUTION OVERVIEW

Although the problem **R-TUM**, (1) – (3), is convex, solving it directly remains challenging because there is no simple characterization of Γ. In this section, we present a general framework of solving **R-TUM** through dual decomposition. The exact distributed algorithm based on DSR will be presented in the next section.

A. Dual Problem Formulation

Let λ_l be the Lagrange multiplier with respect to the constraint $\sum_{f=1}^{F} \bar{\epsilon}_{l,f}^{\omega^+} \le T_l$ in (3), and λ be the vector of all λ_l, $l = 1, 2, \ldots, L$. The Lagrangian of **R-TUM** is then

$$\mathcal{L}(\gamma^{\omega^+}, \lambda) = \sum_{n=1}^{N} \sum_{f=1}^{F} U_{n,f}(\mu_{n,f}^{\omega^+}) - \sum_{l=1}^{L} \lambda_l \left(\sum_{f=1}^{F} \bar{\epsilon}_{l,f}^{\omega^+} - T_l \right), \tag{5}$$

and the dual objective function is

$$\mathcal{D}(\lambda) = \max_{\gamma \in \Gamma} \mathcal{L}(\gamma, \lambda). \tag{6}$$

The dual problem of **R-TUM** is to find a non-negative vector λ that minimizes $\mathcal{D}(\lambda)$.

We first show that strong duality holds for **R-TUM**.

Theorem 1. *Let $\mathcal{P}^*$ be the optimal solution to **R-TUM**, and $\mathcal{D}^* := \min_{\lambda : \lambda_l \ge 0, \forall l} \mathcal{D}(\lambda)$, then $\mathcal{P}^* = \mathcal{D}^*$.*

Proof. The proof is provided in [18]. $\square$

Hence, solving **R-TUM** is equivalent to solving the dual problem, which consists of two steps: First, given a vector λ, we need to find the dual objective function $\mathcal{D}(\lambda)$. Second, we need to find the vector λ that minimizes $\mathcal{D}(\lambda)$.

B. Packet-By-Packet Decomposition for the Dual Objective

We first present an iterative algorithm that finds $\mathcal{D}(\lambda) = \max_{\gamma \in \Gamma} \mathcal{L}(\gamma, \lambda)$ for a given λ using the steepest ascent algorithm. For each stationary policy ω, let γ^ω be defined to be the vector of timely-throughputs and link usages under ω. Then the steepest ascent algorithm constructs a sequence of epoch-wise stationary policies that ultimately converges to the optimal epoch-wise stationary policy. The algorithm proceeds as follows:

1) Set $k \leftarrow 1$
2) Let ω_k^+ be the round-robin epoch-wise stationary policy that follows the sequence $\{\omega_1, \omega_2, \ldots, \omega_k, \omega_1, \omega_2, \ldots, \omega_k, \ldots\}$.
3) Let ω_{k+1} be the stationary policy that maximizes the directional derivative, $\nabla \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \cdot \gamma^{\omega_{k+1}}$.
4) Set $k \leftarrow k+1$ and repeat step 2.

Based on our construction of ω_k^+, we have $\gamma^{\omega_k^+} = \frac{\sum_{j=1}^{k} \gamma^{\omega_j}}{k}$. Therefore $\gamma^{\omega_{k+1}^+} - \gamma^{\omega_k^+} = \frac{\gamma^{\omega_{k+1}} - \gamma^{\omega_k^+}}{k+1}$. Effectively, for each k, our steepest ascent algorithm finds ω_{k+1}^+ that maximizes the directional derivative $\nabla \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \cdot (\gamma^{\omega_{k+1}} - \gamma^{\omega_k^+})$ among all epoch-wise stationary policies with step size $\frac{1}{k+1}$. Following the analysis presented in Boyd et al. [19] Section 9.4.3, it is straightforward to show the following:

Theorem 2. *Under our steepest ascent algorithm, $\mathcal{L}(\gamma^{\omega_k^+}, \lambda)$ converges to $\mathcal{D}(\lambda)$, as $k \to \infty$.*

Notice that the critical step in our steepest ascent policy is to find ω_{k+1} that maximizes $\nabla L(\gamma^{\omega_k^+}, \lambda) \cdot \gamma^{\omega_{k+1}}$. We have

$$
\nabla \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \cdot \gamma^{\omega_{k+1}}
$$
$$
= \sum_{n,f} \frac{\partial}{\partial \mu_{n,f}} \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \mu_{n,f}^{\omega_{k+1}} + \sum_{l,f} \frac{\partial}{\partial \bar{\epsilon}_{l,f}} \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \bar{\epsilon}_{l,f}^{\omega_{k+1}}
$$
$$
= \sum_{f=1}^{F} \left\{ \sum_{n=1}^{N} U'_{n,f}(\mu_{n,f}^{\omega_k^+}) \mu_{n,f}^{\omega_{k+1}} - \sum_{l=1}^{L} \lambda_l \bar{\epsilon}_{l,f}^{\omega_{k+1}} \right\}.
$$

This naturally gives us a flow-by-flow decomposition in the sense that $\nabla \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \cdot \gamma^{\omega_{k+1}}$ can be maximized by maximizing

$$
\sum_{n=1}^{N} U'_{n,f}(\mu_{n,f}^{\omega_k^+}) \mu_{n,f}^{\omega_{k+1}} - \sum_{l=1}^{L} \lambda_l \bar{\epsilon}_{l,f}^{\omega_{k+1}} \tag{7}
$$

for each flow f individually. Moreover, note that, after normalizing with the average packet arrival rate of flow f, $\mu_{n,f}^{\omega_{k+1}}$ is the average delivery per-packet from flow f to node n, and $\bar{\epsilon}_{l,f}^{\omega_{k+1}}$ is the average number of transmissions per packet over link l for flow f.

For each packet i from flow f, let $y_{n,f,i}$ be a random variable representing the event that packet i is successfully delivered to node n within its deadline of d_f. Also, let $z_{l,f,i}$ be the random variable indicating the number times that link l transmits i. Then $\mathbb{E}[y_{n,f,i}]$ is the success probability that packet i is delivered to node n, while $\mathbb{E}[z_{l,f,i}]$

is the expected number of times that link l transmits i. Therefore, from (7), maximizing $\nabla \mathcal{L}(\gamma^{\omega_k^+}, \lambda) \cdot \gamma^{\omega_{k+1}}$ can be achieved by maximizing

$$
\sum_{n=1}^{N} U'_{n,f}(\mu_{n,f}^{\omega_k^+}) \mathbb{E}[y_{n,f,i}] - \sum_{l=1}^{L} \lambda_l \mathbb{E}[z_{l,f,i}] \tag{8}
$$

for each packet i.

We note that such packet-by-packet decomposition allows distributed algorithms for finding the optimal solution since, instead of considering the system state as a whole, each packet only needs to maximize (8) on its own, without considering the states of other packets.

C. Link-by-Link Update for the Dual Problem

After finding $\mathcal{D}(\lambda)$, we now proceed to find the solution to the dual problem, $\min_{\lambda : \lambda_l \geq 0, \forall l} \mathcal{D}(\lambda)$. Our solution is based on the subgradient method. We first find the subgradient of $\mathcal{D}(\lambda)$.

Theorem 3. *Let* $\gamma(\lambda) = [[\mu_{n,f}(\lambda)], [\bar{\epsilon}_{l,f}(\lambda)]] := \arg\max_{\gamma \in \Gamma} \mathcal{L}(\gamma, \lambda)$, *then the L-dimensional vector $[T_l - \sum_{f=1}^{F} \bar{\epsilon}_{l,f}(\lambda)]$ is a subgradient for $\mathcal{D}(\lambda)$.*

Proof. The proof is provided in [18]. $\square$

The subgradient method finds the optimal λ that minimizes $\mathcal{D}(\lambda)$ iteratively. Starting with an arbitrary vector $\lambda(1)$, the subgradient method finds $\lambda(k+1) = [\lambda_l(k+1)]$ by setting

$$
\lambda_l(k+1) = \left[\lambda_l(k) - \beta_k \left(T_l - \sum_{f=1}^{F} \bar{\epsilon}_{l,f}(\lambda(k)) \right) \right]^+, \tag{9}
$$

where $x^+ := \max\{0, x\}$.

Theorem 4. *If the sequence β_k is chosen so that $\beta_k \geq 0, \forall k$, $\sum_{k=1}^{\infty} \beta_k = \infty$, and $\lim_{k \to \infty} \beta_k = 0$, then $\mathcal{D}(\lambda(k)) \to \min_{\lambda : \lambda_l \geq 0, \forall l} \mathcal{D}(\lambda)$, as $k \to \infty$.*

Proof. This is the direct result of Theorem 8.9.2 in [20]. $\square$

Recall that $\sum_{f=1}^{F} \bar{\epsilon}_{l,f}(\lambda(k))$ is the average number of transmissions that link l makes. Therefore, for link l to update λ_l by (9), link l only needs to know its own link constraint and the number of transmissions it makes. Hence, this subgradient method allows for a distributed update of λ_l.

VI. OPTIMIZATION OF DSR

Under DSR, the transmission strategy for a node having a packet i consists of two parts: determining which node to transmit the packet i to, and determining what delegated-set to assign to the receiver. In this section, we discuss the optimal transmission strategy that maximizes (8) under the design of DSR.

Fix a packet i from flow f. For each subset of nodes π, let $\mathcal{L}_\pi$ be the set of links whose transmitter and receiver

are both in π. Also, for each node n, subset of nodes π, and integer $\tau \in [0, d_f]$, define

$$W_f(n, \pi, \tau) =$$
$$\max\left(\sum_{k\in\pi} U'_{k,f}(\mu^{\omega_k^+}_{k,f})\,\mathbb{E}[y_{k,f,i}] - \sum_{l\in\mathcal{L}_\pi} \lambda_l\,\mathbb{E}[z_{l,f,i}]\right) \tag{10}$$

if node n receives the packet i and delegated-set π, and the packet i has τ time slots before meeting its deadline.

By the definition of $W_f(n, \pi, \tau)$, finding the optimal transmission strategy that maximizes (8) is equivalent to finding the value of $W_f(s_f, \{1, 2, \ldots, N\}, d_f)$, as well as the transmission strategy that achieves it.

We use dynamic programming to find $W_f(n, \pi, \tau)$. Suppose node n receives the packet i and delegated-set π, and packet i has τ time slots before meeting its deadline. Also suppose that node n decides to transmit the packet to m and designates the delegated-set π^m to m. If the transmission is successful, then, in the next time slot, node n has a delegated-set of $\pi - \pi^m$, node m has a delegated-set of π^m, and packet i has $\tau - 1$ time slots before its deadline. By the definition of $W_f(\cdot)$, we have, given that the transmission is successful,

$$\max\left(\sum_{k\in\pi} U'_{k,f}(\mu^{\omega_k^+}_{k,f})\,\mathbb{E}[y_{k,f,i}] - \sum_{l\in\mathcal{L}_\pi} \lambda_l\,\mathbb{E}[z_{l,f,i}]\right)$$
$$= W_f(n, \pi - \pi^m, \tau - 1) + W_f(m, \pi^m, \tau - 1) - \lambda_{n\to m}. \tag{11}$$

On the other hand, if the transmission fails, then, in the next time slot, node n still has the delegated-set π and packet i has $\tau - 1$ time slots before its deadline. Given that the transmission fails, we have

$$\max\left(\sum_{k\in\pi} U'_{k,f}(\mu^{\omega_k^+}_{k,f})\,\mathbb{E}[y_{k,f,i}] - \sum_{l\in\mathcal{L}_\pi} \lambda_l\,\mathbb{E}[z_{l,f,i}]\right)$$
$$= W_f(n, \pi, \tau - 1) - \lambda_{n\to m}. \tag{12}$$

Since each transmission from n to m succeeds with probability $P_{n\to m}$, we have, given that n transmits packet i and assigns delegated-set π^m to m,

$$\max\left(\sum_{k\in\pi} U'_{k,f}(\mu^{\omega_k^+}_{k,f})\,\mathbb{E}[y_{k,f,i}] - \sum_{l\in\mathcal{L}_\pi} \lambda_l\,\mathbb{E}[z_{l,f,i}]\right)$$
$$= P_{n\to m} \times (11) + (1 - P_{n\to m}) \times (12). \tag{13}$$

Based on the above analysis, we can write down the following iterative equation:

$$W_f(n, \pi, \tau) = \max\{W_f(n, \pi, \tau - 1),$$
$$\max_{m, \pi^m : m\in\pi^m, \pi^m\subset\pi}[P_{n\to m}(W_f(n, \pi - \pi^m, \tau - 1)$$
$$+ W_f(m, \pi^m, \tau - 1)) + (1 - P_{n\to m})W_f(n, \pi, \tau - 1)$$
$$- \lambda_{n\to m}]\}, \tag{14}$$

with boundary condition

$$W_f(n, \pi, 0) = r_{i,n} = U'_{n,f}(\mu_{n,f}), \tag{15}$$

where the term $W_f(n, \pi, \tau - 1)$ in (14) represents the case when n does not transmit the packet at all. Eq. (14) and (15) allows a dynamic programming algorithm to find $W_f(n, \pi, \tau)$ for all f, n, π, and τ. As we will show in Section VII, our algorithm can be easily carried out in medium-sized networks.

A. Index-DSR for Per-Time-Slot Link Constraint

The Dynamic Program in (14) can be directly combined with the dual decomposition in Section V to achieve the optimal solution of **R-TUM** problem under DSR. In this section, we further propose an index policy that satisfies the per-time-slot link utilization constraint $\sum_{i,v} \epsilon_{i,v,l}(t) \leq T_l$, for all t, of the original **TUM** problem. The index-DSR policy would be to transmit the maximum number of packets among all possible packets to be transmitted so that the per-time-slot link constraint is not violated.

We make several changes to the dynamic program and the dual decomposition technique. First, we change the iterative equation (14) to

$$W_f(n, \pi, \tau) =$$
$$\max_{m, \pi^m : m\in\pi^m, \pi^m\subset\pi}[P_{n\to m}(W_f(n, \pi - \pi^m, \tau - 1)$$
$$+ W_f(m, \pi^m, \tau - 1)) + (1 - P_{n\to m})W_f(n, \pi, \tau - 1)$$
$$- \lambda_{n\to m}]\}, \tag{16}$$

as long as there is a link from n to another node in π, and

$$W_f(n, \pi, \tau) = W_f(n, \pi, \tau - 1), \tag{17}$$

otherwise. In other words, we force each node n to find a link to transmit each packet. We also define $m^*(n, \pi, \tau)$ and $\pi^{m*}(n, \pi, \tau)$ as the optimal m and π^m that achieves $W_f(n, \pi, \tau)$. We note that, since we now force each node n to find a link to transmit each packet, it is possible that $W_f(n, \pi, \tau)$ is negative for some (n, π, τ).

Second, in each time slot t and for each link $n \to m$, we find all packets possessed by n with delegated-set π, τ slots until their respective deadlines, and $m^*(n, \pi, \tau) = m$. We sort these packets in descending order of $W_f(n, \pi, \tau)$, and let $\epsilon'_{n\to m}(t)$ be the number of these packets with $W_f(n, \pi, \tau) > 0$. In other words, $\epsilon'_{n\to m}(t)$ is the number of packets whose optimal strategy yields a positive return by transmitting over the link $n \to m$. After sorting these packets, link $n \to m$ simply transmit the first $T_{n\to m}$ packets. Finally, the price of each link is updated by (9).

VII. SIMULATION RESULTS

In this section, we present simulation results that compare the performance of our policy against a policy proposed in [21] called Universal Max-Weight (UMW), and a policy proposed by Sinha, Paschos, and Modiano in [13] that we call SPM. We first provide a brief description of these two policies.
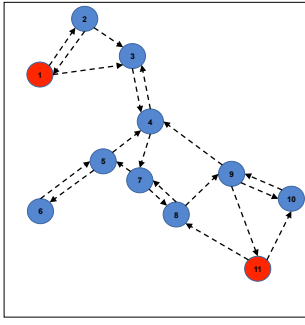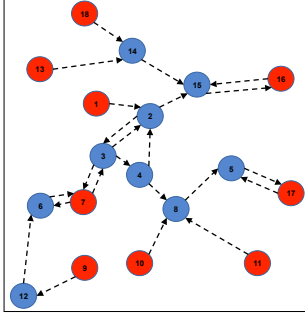
Fig. 2. Scenario 1: 11−node network

Fig. 3. Scenario 1: Linear Utility Function
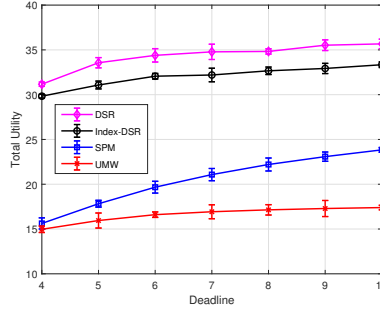

Fig. 4. Scenario 1: Logarithmic Utility Function
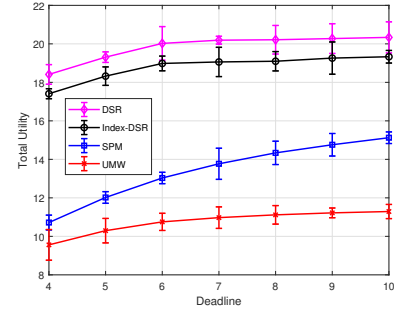

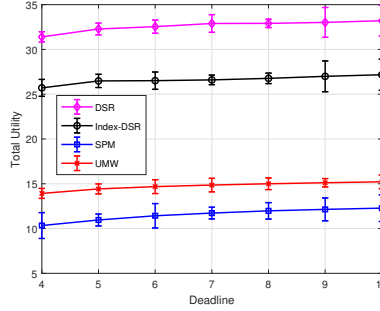Fig. 5. Scenario 2: 18−node network
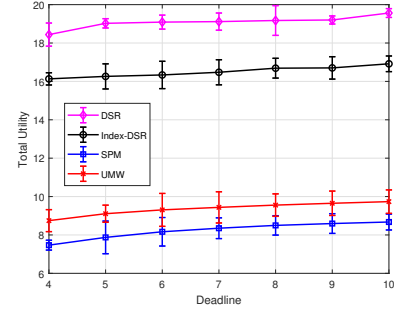

Fig. 6. Scenario 2: Linear Utility Function

Fig. 7. Scenario 2: Logarithmic Utility Function

A. Overview of UMW and SPM

The UMW policy solves the problem of throughput-optimal packet dissemination in a network with arbitrary topology with unicast, multicast and broadcast traffic. In both the centralized and distributed versions of UMW, the route of each packet is decided at the origin. This route is a weighted tree that is constructed using the edge weights at time of decision at the origin. Hence, if the route of the packet turns to be inappropriate during packet dissemination, it cannot be modified. This policy also has a heuristic distributed, but compare against the centralized version, which has better performance than the distributed one.

The SPM policy is designed for throughput optimal broadcast. SPM is a virtual-queue based algorithm, where virtual-queues are defined for subsets of nodes. These virtual queues keep track of a kind of backpressure, while accounting for the fact that packets are duplicated in the broadcast regime. Each slot is sub-divided into L minislots, where L is the number of links in the network, and a random link is activated in each mini-slot. Here, a packet may be retransmitted multiple times over the mini-slots comprising a slot (i.e., it could potentially reach all nodes in just one slot). To ensure consistency with the slot model, we modify this algorithm to only allow packet state updates at each slot, rather than at each mini-slot.

B. Simulation Settings and Results

In this study, we consider two different simulation scenarios motivated by designs for IAB network deploy-ments [1], [2]. Here, we have two kinds of nodes, namely, (i) gateway nodes with fiber drops (shown in red), and (ii) wireless-only nodes with mm-wave backhaul (shown as blue nodes). We assume that gateways communicate reliably between each other with zero latency, since they are connected to the same backend switch (consistent with IAB architecture). The two scenarios represent different levels of gateway availability. The first scenario is a an $11-$node network with 2 fiber drops as in figure (2), while the second scenario is an $18-$ node IAB network with 9 fiber drops as shown in figure (5). Hence, Scenario 1 is illustrative performance in a network with multiple wireless hops, whereas Scenario 2 illustrates performance in a more densely connected network.

In both scenarios, there are two broadcast flows. One of the flows originates at a fiber-connected gateway node, and the other one from a wireless-only node. For each link l, P_l is randomly chosen from $[0.5, 1.0]$, and T_l is randomly chosen from $[1, 5]$. Each flow generates packets according to a Poisson random process, where source node of flow 1 has a mean arrival rate of 1.5 packets per time slot, and source node of second flow has a mean arrival rate of 2 packets per time slot. Since UWM and SPM only aim to maximize throughput, we first consider a linear utility function $U_{n,f}(\mu_{n,f}) = \mu_{n,f}$ to make a fair comparison. In this case, the total utility of the system is the same as the total timely-throughputs. In a second case, we also consider a logarithmic utility function $U_{n,f}(\mu_{n,f}) = \log(\mu_{n,f}+1)$, which models the idea that the utility of the end user might be a non-negative,

concave and increasing function of timely throughput. We assume that the two flows have the same deadline of D time slots, and vary D from 4 to 10. We test four the optimal DSR protocol (for the relaxed problem), the Index-DSR protocol, the UWM policy, and the SPM policy.

The simulation results for the linear utility function and the logarithmic utility function for scenario in figure (2) are shown in figures (3) and (4), respectively. The performance of DSR is an upper bound, since it is the optimal solution under a relaxed constraint. The Index-DSR protocol outperforms UWM, possibly because of more dynamic routing of each packet under Index-DSR. This also shows that UMW might be providing bursty service to nodes, since deadlines are often violated and packets are dropped, leading to poor throughput. The Index-DSR policy outperforms SPM in all cases despite the assumption that SPM can compute the reachable subgraph for each packet instantly.

The results for second IAB scenario, depicted in (5) shows similar results in terms performance of DSR-based algorithms for much the same reasons specified above. However, results of figures (6) and (7) shows that UMW has better performance than SPM, unlike the results obtained in (3) and (7). This result appears to be due to the density of the network. The UMW policy manages to deliver more unexpired packets to the destination since it has to traverse fewer hops. SPM is also handicapped by the fact that we force it to obey a slot-by-slot state update model like all the other protocols (although we do allow it to utilize its minislot-based transmission model). Ultimately, these results demonstrate the efficiency and flexibility of the DSR protocol.

VIII. CONCLUSION

We studied the problem of broadcasting real-time flows with hard per-packet deadlines in a multi-hop wireless network. This problem is computationally complex due to the need to solve an MDP over the network graph. We relax the problem using average link utilization constraints, and come up with a novel decomposition approach that enables its solution in a distributed fashion. We propose the DSR algorithm that maximizes the total timely-utility. The algorithm has a low complexity, and has a really low coordination overhead. We also develop a simple *index* policy based on DSR that is able to meet hard link utilization constraints. We simulate the variants of the algorithm, comparing against several recent throughout optimal algorithms . In all cases, DSR and the index policy have a better performance in terms of total timely-utility. We conclude that throughput and delay optimality are fundamentally different, but simple near-optimal solutions are possible in the delay-constrained case.

REFERENCES

[1] M. N. Islam, N. Abedini, G. Hampel, S. Subramanian, and J. Li, "Investigation of performance in integrated access and backhaul networks," *arXiv preprint arXiv:1804.00312*, 2018.

[2] M. N. Islam, S. Subramanian, and A. Sampath, "Integrated access backhaul in millimeter wave networks," in *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pp. 1–6, IEEE, 2017.

[3] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for reliable multicast in multi-hop ad hoc networks," in *In Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM)*, 1999.

[4] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson, "A simple protocol for multicast and broadcast in mobile ad hoc networks," in *In Internet Draft, draft-ietf-nanet-simple-mbcast-00.txt*, June 2001.

[5] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *In Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151 – 162, 1999.

[6] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *in Proceedings of 4th ACM international symposium on Mobile ad hoc networking and computing*, pp. 222 – 232, 2003.

[7] S. C. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, "Minimum-latency broadcast scheduling in wireless ad hoc networks," in *Proc. of IEEE INFOCOM*, 2007.

[8] S. Sarkar and L. Tassiulas, "A framework for routing and congestion control for multicast information flows," *IEEE Transactions on Information Theory*, vol. 48, no. 10, pp. 2690 – 2708, 2002.

[9] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *In Proc. 43rd Annual Allerton Conference on Communication Control, and Computing*, 2005.

[10] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multihop multicast in wireless mesh networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, 2006.

[11] S. Zhang, M. Chen, Z. Li, and L. Huang, "Optimal distributed broadcasting with per-neighbor queues in acyclic overlay networks with arbitrary underlay capacity constraints," in *In Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pp. 814 – 818, 2013.

[12] 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*, pp. 21 – 30, July 2016.

[13] 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*, pp. 51 – 60, July 2016.

[14] H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-aware cross-layer design for network utility maximization in multi-hop networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, pp. 951 – 959, 2011.

[15] Z. Mao, C. Koksal, and N. Shroff, "Online packet scheduling with hard deadlines in multihop communication networks," in *Proc. of IEEE INFOCOM*, pp. 2463 – 2471, 2013.

[16] R. Li and A. Eryilmaz, "Scheduling end-to-end deadline-constrained traffic with reliability requirements in multi-hop networks," *Selected Areas in Communications, IEEE Journal on*, vol. 20, pp. 1649 – 1662, Otc 2012.

[17] R. Singh and P. Kumar, "Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 127–142, 2019.

[18] Aria HasanzadeZonuzy, I-Hong Hou and Srinivas Shakkottai, "Broadcasting Real-Time Flows in Integrated Backhaul and Access 5G Networks." https://www.dropbox.com/s/zsiz4kio9i4r9mg/e2e_broadcast.pdf?dl=0, 2019.

[19] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[21] A. Sinha and E. Modiano, "Optimal control for generalized network-flow problems," *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 1, pp. 506–519, 2018.