

Optimal multicast throughput in random access networks of general topology

Brooke Shrader

Dept. of Electrical and Computer Engineering
University of Maryland
College Park, MD 20742
Email: bshrader@umd.edu

Matthew Andrews

Bell Laboratories
Alcatel-Lucent
Murray Hill, NJ 07974
Email: andrews@research.bell-labs.com

Abstract—We consider multicast transmissions in a wireless ad hoc network where nodes randomly compete for access to a shared channel. Our goal is in optimizing a weighted proportional fairness objective of the network throughput. We consider two different forms of multicast: non-guaranteed and guaranteed. In both cases, we characterize the multicast throughput and provide schemes to compute the optimal channel access probabilities.

I. INTRODUCTION

A number of applications, including disaster recovery, military, and conferencing applications, require that multicast transmissions be carried out in a distributed fashion in a wireless network. In this work we examine multicast transmissions at the medium access control (MAC) layer and focus on the use of a slotted random access protocol, which is a simple, distributed protocol for coordinating access to a shared channel. We consider a network of general connection topology and propose schemes to assign the node access probabilities in order to optimize a weighted proportional fairness objective function.

The model we consider is as follows. The network consists of a finite set \mathcal{N} of nodes and each node n serves multiple multicast flows. Source n together with the one-hop neighbors in a multicast flow emanating from that source constitute a *multicast tree* and the set of receivers in the tree is denoted \mathcal{D}_{nm} , where m indexes the multicast tree. Time is slotted, and in each time slot, node n accesses the channel with probability p_n and chooses to transmit on multicast tree m with probability p_{nm} . Thus $p_n = \sum_m p_{nm}$. A transmission by node n interferes with reception at all nodes in a set \mathcal{N}_n ; interference at the receiver causes destruction of any transmitted

packets intended for that receiver. Interference is not assumed to be symmetric - for nodes $n, k \in \mathcal{N}$, $k \in \mathcal{N}_n$ does not necessarily imply that $n \in \mathcal{N}_k$. We consider two different multicast scenarios: non-guaranteed and guaranteed multicast. In both cases, we characterize the network throughput and find the access probabilities that maximize the proportional fairness objective. Our results are summarized below.

- **Non-guaranteed multicast:** We assign a distinct link weight $w_{nmd} > 0$ for all destinations $d \in \mathcal{D}_{nm}$. The weighted proportional fairness problem is

$$\max_{\boldsymbol{\mu}_{\mathcal{T}}} \sum_{(n, \mathcal{D}_{nm}, d)} w_{nmd} \log \mu_{nmd} \quad (1)$$

where μ_{nmd} is the throughput obtained by receiver d on tree (n, \mathcal{D}_{nm}) , i.e. the number of packets successfully received by receiver d on tree (n, \mathcal{D}_{nm}) per unit time, and $\boldsymbol{\mu}_{\mathcal{T}} = \{\mu_{nmd}\}$ is the vector of μ_{nmd} for all links in the network. This approach is of interest for applications in which it is beneficial, but not required, for each destination in the multicast tree to receive the transmitted packets.

- **Guaranteed multicast:** For applications that require that all destinations in the multicast tree *must* receive the transmitted packets, we assign weight $w_{nm} > 0$ to the multicast tree (n, \mathcal{D}_{nm}) and are interested in the problem

$$\max_{\boldsymbol{\mu}_{\mathcal{T}}} \sum_{(n, \mathcal{D}_{nm})} w_{nm} \log \mu_{nm} \quad (2)$$

where μ_{nm} is the multicast throughput on tree (n, \mathcal{D}_{nm}) , i.e. the number of packets that have been successfully received by all receivers on the tree per unit time, and $\boldsymbol{\mu}_{\mathcal{T}} = \{\mu_{nm}\}$ is the vector of μ_{nm} for all trees in the network. We emphasize that although all receivers need to receive the packet in

order to get credit, they do not have to receive it in the same timeslot. We characterize μ_{nm} for a retransmission strategy as well as for a packet (fountain) coding strategy. We then recast the weighted proportional fairness problem by introducing the constraint $\mu_{nm} \leq \mu_{nmd}, \forall d \in \mathcal{D}_{nm}$ and provide an algorithm that converges to the optimal set of access probabilities p_{nm} .

Previous work has considered optimal throughput allocation for *unicast* transmission in random access. In works by Kar et al. [1] and Gupta and Stolyar [2], the access probabilities are assigned in order to optimize a weighted proportional fairness objective function of the single-hop throughput. The access probabilities in this case can be computed using only the link weights in the local neighborhood (2-hop neighbors). In [3], Wang and Kar propose distributed algorithms to solve the weighted proportional fairness problem for end-to-end throughput. A family of objective functions which encapsulates proportional fairness, max-min fairness, and other types of fairness is presented by Mo and Walrand in [4]; this family of objective functions is applied to a random access network by Lee et al. in [5].

There has also been previous work in designing efficient multicast transmission strategies. In [6], Chaporkar and Sarkar propose a strategy in which the number of receivers that are available for reception is compared to a threshold in order to determine whether the source should transmit. The same authors study a multicast transmission strategy aimed at minimizing delay in [7]. In the models considered in [6], [7], the ability of a node to receive a transmitted packet is determined by some exogenous process, i.e., they are given as part of the problem input; in contrast, in our work, the ability of a node to receive depends on the access probabilities of neighboring nodes as determined by our algorithm. In [8], Kar et al. study the end-to-end multicast rate control problem under the assumption that different nodes in the multicast group can receive at different rates, which is similar to the non-guaranteed multicast we consider here. The work in [8] treats the multicast problem for a wireline network, whereas in the present work we consider wireless transmission.

II. THE MODEL

An example of the setting we consider is shown in Fig. 2. The network consists of a finite set \mathcal{N} of nodes. Node $n \in \mathcal{N}$ has traffic to send to some subset of its neighbors. Let $\mathcal{D}_n \subseteq \mathcal{N}$ denote the set of nodes for which n has some traffic to send. The set \mathcal{D}_n consists of

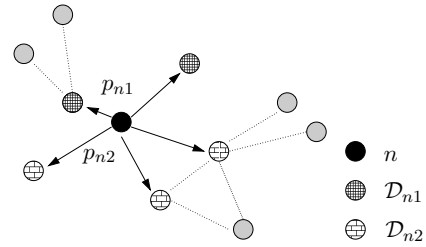


Fig. 1. An example of the multicast setting we consider in this work.

subsets $\mathcal{D}_{nm}, m \in \mathcal{D}_n$, for which node n intends to send the same traffic, i.e., node n will *multicast* traffic to all nodes in the set \mathcal{D}_{nm} where $|\mathcal{D}_{nm}| \geq 1$. This model can account for a mix of unicast and multicast traffic since unicast would correspond to $|\mathcal{D}_{nm}| = 1$.

We assume that time is slotted and in each time slot, node n can attempt transmission of a packet to (all nodes in) one of the sets \mathcal{D}_{nm} . We say that node n makes a transmission attempt on multicast tree (n, \mathcal{D}_{nm}) . (This is a tree of depth 1.) Denote by

$$\mathcal{T} = \{(n, \mathcal{D}_{nm}) : n \in \mathcal{N}, \mathcal{D}_{nm} \subseteq \mathcal{D}_n\} \quad (3)$$

the set of all (depth-1) multicast trees in the network. Additionally, let $d \in \mathcal{D}_{nm}$ denote a leaf in the tree (n, \mathcal{D}_{nm}) and (n, \mathcal{D}_{nm}, d) denote the link between n and d in the tree. The set \mathcal{I} denotes the set of all multicast links in the network.

$$\mathcal{I} = \{(n, \mathcal{D}_{nm}, d) : n \in \mathcal{N}, \mathcal{D}_{nm} \subseteq \mathcal{D}_n, d \in \mathcal{D}_{nm}\} \quad (4)$$

We make the following assumptions with regard to transmissions and the channel. (I) A node n can transmit on at most one multicast tree (n, \mathcal{D}_{nm}) in each time slot. (II) A node can receive at most 1 packet per slot. (III) Any transmission attempt by node n will interfere with and destroy any attempt to receive a packet at any of the nodes in a set \mathcal{N}_n where $\mathcal{D}_n \subseteq \mathcal{N}_n$ and $n \in \mathcal{N}_n$. Nodes in the network make use of a random access strategy as follows. In each time slot, node n transmits with probability p_n . When n transmits, it chooses a particular multicast tree to transmit on from among $(n, \mathcal{D}_{nm}), \mathcal{D}_{nm} \subseteq \mathcal{D}_n$, randomly with probability p_{nm}/p_n where $\sum_{m \in \mathcal{D}_n} p_{nm} = p_n$. We assume throughout that each multicast tree $(n, \mathcal{D}_{nm}) \in \mathcal{T}$ in the network has an infinite backlog of packets awaiting transmission on the tree.

III. NON-GUARANTEED MULTICAST

We first focus our attention on the optimization problem in Eqn. (1). Note that for a transmitting node n and

receiving node d , the weight w_{nmd} and throughput μ_{nmd} may take different values depending on the multicast tree (or multicast flow) of interest, i.e., links (n, \mathcal{D}_{nm}, d) and (n, \mathcal{D}_{nl}, d) , $m \neq l$, are distinct, with $w_{nmd} \neq w_{nld}$. For a given link (n, \mathcal{D}_{nm}, d) it can be shown that μ_{nmd} is maximized by a policy in which each packet is transmitted on (n, \mathcal{D}_{nm}) *only once*. Once the packet has been transmitted on (n, \mathcal{D}_{nm}) , it is removed from memory at n . Then the multicast transmission process on (n, \mathcal{D}_{nm}) consists of only a single transmission and the average throughput is given by

$$\mu_{nmd} = p_{nm} \prod_{k:d \in \mathcal{N}_k, k \neq n} (1 - p_k). \quad (5)$$

Let \mathcal{L}_n denote the set of links (l, \mathcal{D}_{lk}, j) , $j \in \mathcal{D}_{lk}$, that either originate at n or are such that a transmission by n causes interference and destroys that on (l, \mathcal{D}_{lk}, j) .

$$\mathcal{L}_n = \{(l, \mathcal{D}_{lk}, j) : j \in \mathcal{D}_{lk}, j \in \mathcal{N}_n\} \quad (6)$$

Define further \mathcal{L}_n^- as the set of links in \mathcal{L}_n that do not originate at n .

$$\mathcal{L}_n^- = \{(l, \mathcal{D}_{lk}, j) \in \mathcal{L}_n : l \neq n\} \quad (7)$$

The following theorem describes the optimal throughput allocation for non-guaranteed multicast. Our analysis is based on techniques used in [2].

Theorem 1. *For arbitrary sets of positive real-valued weights $\{w_{nmd}, (n, \mathcal{D}_{nm}, d) \in \mathcal{I}\}$, there exists a unique set of access probabilities that maximize the function*

$$\sum_{(n, \mathcal{D}_{nm}, d) \in \mathcal{I}} w_{nmd} \log \mu_{nmd}. \quad (8)$$

The optimal probabilities are given by

$$p_{nm} = \frac{\sum_{d \in \mathcal{D}_{nm}} w_{nmd}}{\sum_{(l, \mathcal{D}_{lk}, j) \in \mathcal{L}_n} w_{lkj}}. \quad (9)$$

Proof: The proof is similar to the proof of [2, Theorem 1]. In addition, we note that

$$\mathcal{L}_n = \mathcal{L}_n^- \cup \left\{ \bigcup_{m \in \mathcal{D}_n} \left\{ \bigcup_{d \in \mathcal{D}_{nm}} (n, \mathcal{D}_{nm}, d) \right\} \right\} \quad (10)$$

which gives rise to the sum $\sum_{d \in \mathcal{D}_{nm}} w_{nmd}$ in the numerator of (9). ■

IV. GUARANTEED MULTICAST

We now address the weighted proportional fairness problem in Eqn. (2). Here μ_{nm} denotes the throughput for the tree, or the number of packets per slot that can be successfully transmitted to *all* recipients on the tree. The multicast throughput μ_{nm} is in general difficult to describe. The difficulty arises in that the intended recipients on (n, \mathcal{D}_{nm}) have distinct sets of neighboring, interfering nodes and as such, the intended recipients have distinct probabilities of successfully receiving a packet transmitted by n . Node n might employ different strategies to ensure successful reception $\forall d \in \mathcal{D}_{nm}$, e.g., node n might retransmit the packet on (n, \mathcal{D}_{nm}) as long as the packet has not been received by some $d \in \mathcal{D}_{nm}$. Regardless of the transmission strategy that node n employs, due to the non-uniformity of the channels on the links (n, \mathcal{D}_{nm}, d) , $d \in \mathcal{D}_{nm}$, the process by which node n transmits on (n, \mathcal{D}_{nm}) to ensure successful reception $\forall d \in \mathcal{D}_{nm}$ is a process with memory.

In this section, we characterize μ_{nm} for a retransmission strategy as well as for a packet (fountain) coding strategy. We then recast the weighted proportional fairness problem by introducing the constraint $\mu_{nm} \leq \mu_{nmd}$, $\forall d \in \mathcal{D}_{nm}$ and provide an algorithm that converges to the optimal set of access probabilities p_{nm} .

A. Throughput: retransmission strategy

We now consider a strategy whereby node n repeatedly transmits a packet on the multicast tree (n, \mathcal{D}_{nm}) until it has received a feedback acknowledgement of successful reception of the packet from all intended recipients in the tree, i.e., $\forall d \in \mathcal{D}_{nm}$. For destination $d \in \mathcal{D}_{nm}$, the probability of successful reception of a packet in any given time slot is equal to μ_{nmd} , the average unicast throughput on link (n, \mathcal{D}_{nm}, d) . The transmission time to destination d , or the number of slots that elapse from the time a packet is first available for transmission on tree (n, \mathcal{D}_{nm}) until the end of the slot in which destination d successfully receives the packet, is a geometric random variable X_d with parameter μ_{nmd} , where $\Pr(X_d > 0) = 1$. (Equivalently, X_d is the number of independent coin tosses needed up to and including the first toss that results in a heads, where heads occurs in each toss with probability μ_{nmd} .) The total transmission time, or the number of slots needed until all intended recipients on (n, \mathcal{D}_{nm}) receive the packet, has the expected value $E[\max_{d \in \mathcal{D}_{nm}} X_d]$. As noted earlier, the transmission process described above is a process with memory. Additionally, the random variables $X_d, d \in \mathcal{D}_{nm}$, may be correlated since the

intended recipients may have overlapping sets of interfering nodes. In the following lemma, we bound the throughput for this retransmission strategy.

Lemma 1. *The guaranteed multicast throughput μ_{nm}^R for the retransmission strategy is bounded as*

$$\frac{\mu_{nmd}^{\min}}{(1 + \log |\mathcal{D}_{nm}|) \left(1 + \frac{2}{e \left(1 - \frac{1}{e^{|\mathcal{D}_{nm}|}}\right)^2}\right)} \leq \mu_{nm}^R \leq \mu_{nmd}^{\min} \quad (11)$$

where

$$\mu_{nmd}^{\min} = p_{nm} \min_{d \in \mathcal{D}_{nm}} \prod_{k: d \in N_k, k \neq n} (1 - p_k) \quad (12)$$

Proof: The result follows by bounding the expected total transmission time $E[\max_{d \in \mathcal{D}_{nm}} X_d]$ and by noting that $\mu_{nm}^R = 1/E[\max_{d \in \mathcal{D}_{nm}} X_d]$. The upper bound in (11) can be shown as follows.

$$E[\max_d X_d] \stackrel{(a)}{\geq} \max_d E[X_d] = \max_d \frac{1}{\mu_{nmd}} = \frac{1}{\mu_{nmd}^{\min}}$$

where (a) holds by Jensen's inequality since the max function is convex. To show the lower bound in (11), let c denote a fixed constant. The following sequence of inequalities holds for any $c > 1$.

$$\begin{aligned} E[\max_d X_d] &\leq \sum_{i=1}^{\infty} \frac{ic}{\mu_{nmd}^{\min}} \Pr\left(\frac{(i-1)c}{\mu_{nmd}^{\min}} < \max_d X_d \leq \frac{ic}{\mu_{nmd}^{\min}}\right) \\ &\leq \sum_{i=1}^{\infty} \frac{ic}{\mu_{nmd}^{\min}} \Pr\left(\frac{(i-1)c}{\mu_{nmd}^{\min}} < \max_d X_d \leq \infty\right) \\ &\stackrel{(b)}{\leq} \frac{c}{\mu_{nmd}^{\min}} + \sum_{i=2}^{\infty} \frac{ic}{\mu_{nmd}^{\min}} \sum_d (1 - \mu_{nmd})^{\frac{(i-1)c}{\mu_{nmd}^{\min}}} \\ &\stackrel{(c)}{\leq} \frac{c}{\mu_{nmd}^{\min}} \left(1 + |\mathcal{D}_{nm}| \sum_{i=1}^{\infty} (i+1)e^{-ci}\right) \\ &= \frac{c}{\mu_{nmd}^{\min}} \left(1 + \frac{2|\mathcal{D}_{nm}|e^{-c} - |\mathcal{D}_{nm}|e^{-2c}}{(1 - e^{-c})^2}\right) \end{aligned}$$

where (b) holds by the union bound and by using the fact that X_d is geometrically distributed and (c) holds since $(1 - \mu_{nmd})^{(i-1)c/\mu_{nmd}^{\min}} \leq e^{-(i-1)c\mu_{nmd}/\mu_{nmd}^{\min}} \leq e^{-(i-1)c}$. By taking $c = 1 + \log |\mathcal{D}_{nm}|$ we obtain

$$E[\max_d X_d] \leq \frac{1}{\mu_{nmd}^{\min}} (1 + \log |\mathcal{D}_{nm}|) \left(1 + \frac{2}{e \left(1 - \frac{1}{e^{|\mathcal{D}_{nm}|}}\right)^2}\right)$$

This provides the lower bound in (11). ■

Lemma 1 shows that when using the retransmission strategy, the guaranteed multicast throughput on tree

(n, \mathcal{D}_{nm}) is upper bounded by the (unicast) throughput to the destination $d \in \mathcal{D}_{nm}$ that has the *smallest* (unicast) throughput among all destinations in the multicast tree.

B. Throughput: coded transmission strategy

We now consider a strategy whereby node n performs coding over groups of packets awaiting transmission on multicast tree (n, \mathcal{D}_{nm}) . Let K denote the number of packets involved in encoding, or the number of inputs to the encoder. The coding strategy works as follows. For each transmission attempt on (n, \mathcal{D}_{nm}) , the encoder at node n samples from a *degree distribution* to obtain a value w between 1 and K . The encoder then selects w packets randomly and uniformly from among the K input packets and forms the sum (modulo-2) of these w packets, which is the output of the encoder. The encoder output is then transmitted on the multicast tree (n, \mathcal{D}_{nm}) . We assume that the coefficients of the random sum are transmitted in the header of the packet; these coefficients will be needed to perform decoding.

In the next transmission attempt on (n, \mathcal{D}_{nm}) , the encoder follows the same procedure to independently and randomly form a new output for transmission on the tree. This process continues as the intended recipients on (n, \mathcal{D}_{nm}) collect coded versions of the K packets. Once a destination $d \in \mathcal{D}_{nm}$ has received N encoder outputs, where $N \gtrsim K$, and is able to decode the original K packets, it sends a feedback acknowledgement to node n . As soon as node n has collected acknowledgements from all intended recipients in \mathcal{D}_{nm} , the transmission process is complete and node n can commence encoding and transmission of another group of K packets awaiting transmission on (n, \mathcal{D}_{nm}) . The coding strategy described here is a general description of Fountain coding.

For an appropriately designed degree distribution, the value of N can be made arbitrarily close to K and decoding can be performed by belief propagation with low computational complexity. The LT-codes invented by Luby in [9] can be implemented through use of the Robust Soliton Distribution, for which $N = K + O(\sqrt{K}(\log K)^2)$ ensures that belief propagation decoding can be performed with error at most k^{-c} , $c > 0$ and decoding complexity on the order of $K \log K$. The Raptor codes described in [10], which were proposed by Shokrollahi and involve concatenating an error-correcting pre-code with an LT-code, provide better performance in the sense that N can be made even closer to K with small error probability and low-complexity decoding. For the purpose of describing the multicast throughput, we do not assume the use of a

particular Fountain code. More generally, we assume that for a given error probability and decoding complexity, there exists a deterministic value N which is arbitrarily close to (but slightly larger than) K .

For destination $d \in \mathcal{D}_{nm}$, the probability of successful reception of an encoder output in any given time slot is μ_{nmd} . The transmission time to destination d , or the number of slots that elapse from the time a group of K packets is first available for transmission on tree (n, \mathcal{D}_{nm}) until the end of the slot in which destination d successfully receives the N^{th} encoder output, is given by $Y_d = X_{d,1} + X_{d,2} + \dots + X_{d,N}$, where $X_{d,i}$, $1 \leq i \leq N$ are independently, geometrically distributed with parameter μ_{nmd} . The geometric distribution of $X_{d,i}$ does not imply that a particular encoder output is retransmitted until received; it merely indicates that if a new encoder output is generated at each transmission attempt, then the number of slots that elapse until one encoder output is received is geometrically distributed. (Equivalently, if instead of tossing a single coin, we have a collection of coins with identical bias and choose a different coin for each toss, then the number of independent tosses until the first heads appears is geometric.) The total transmission time, or the number of slots needed for all destinations on (n, \mathcal{D}_{nm}) to receive N encoder outputs is $E[\max_{d \in \mathcal{D}_{nm}} Y_d]$.

Lemma 2. *The guaranteed multicast throughput μ_{nm}^C for the coded transmission strategy is bounded as*

$$\mu_{nm}^C \leq \mu_{nmd}^{\min} \quad (13)$$

where the bound holds with equality in the limit as $K \rightarrow \infty$.

Proof: We again make use of Jensen's inequality to show that

$$E[\max_{d \in \mathcal{D}_{nm}} Y_d] \geq \max_d E[Y_d] = \max_d \frac{N}{\mu_{nmd}} = \frac{N}{\mu_{nmd}^{\min}}. \quad (14)$$

By using $\mu_{nm}^C = K/E[\max_d Y_d]$ and by the assumption that the Fountain code allows for N to be made arbitrarily close to K , the upper bound in the lemma follows.

To show when the upper bound holds with equality, we argue as follows. Let \mathbf{Y}_d denote the vector of transmission times for $d \in \mathcal{D}_{nm}$ and let \mathbf{Y}_d^a and \mathbf{Y}_d^b denote two realizations of the vector \mathbf{Y}_d . The convexity of the max function means that for any realizations $\mathbf{Y}_d^a, \mathbf{Y}_d^b$

$$\max(\theta \mathbf{Y}_d^a + (1 - \theta) \mathbf{Y}_d^b) \leq \theta \max \mathbf{Y}_d^a + (1 - \theta) \max \mathbf{Y}_d^b \quad (15)$$

where $0 \leq \theta \leq 1$. As $K \rightarrow \infty$, by the strong law of large numbers, $Y_d \rightarrow N/q_d$ with probability 1 and (15) holds with equality. This is true even if the values of $\mu_{nmd}, d \in \mathcal{D}_{nm}$ are not distinct as long as ties are broken (i.e., a unique destination with the worst channel is selected) by an arbitrary but fixed rule. Since (15) holds with equality, Jensen's also holds with equality and the upper bound on the throughput is tight. ■

Lemma 2 shows that as with the retransmission strategy, the guaranteed multicast throughput for the coded transmission strategy is upper bounded by the smallest unicast throughput among all destinations on the tree (n, \mathcal{D}_{nm}) . However, unlike the retransmission strategy, the coded transmission strategy can reach the upper bound on the throughput, but at the cost of infinite delay.

C. Weighted proportional fairness for guaranteed multicast

We now return to the problem of maximizing the weighted proportional fairness for guaranteed multicast. We can formulate the problem as maximizing the weighted proportional fairness subject to a constraint on the throughput μ_{nm} given by Lemmas 1 and 2. Our constraint is

$$\mu_{nm} \leq \mu_{nmd}^{\min} \iff \mu_{nm} \leq \mu_{nmd}, \quad \forall d \in \mathcal{D}_{nm} \quad (16)$$

where, as shown in Lemma 2, $\mu_{nm} = \mu_{nmd}^{\min}$ is achievable by making use of the coded transmission policy. Our weighted proportional fairness problem \mathbf{P} is stated below.

$$\begin{aligned} \mathbf{P} : \quad & \max \sum_{(n, \mathcal{D}_{nm}) \in \mathcal{T}} w_{nm} \log \mu_{nm} \\ & \text{s.t.} \quad \mu_{nm} \leq \mu_{nmd}, \quad \forall d \in \mathcal{D}_{nm}, \forall (n, \mathcal{D}_{nm}) \in \mathcal{T} \\ & \quad \mu_{nmd} = c_{nmd}(\mathbf{p}), \quad \forall (n, \mathcal{D}_{nm}, d) \in \mathcal{I} \\ & \quad 0 \leq p_{nm} \leq 1, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{T} \\ & \quad p_n \leq 1, \quad \forall n \in \mathcal{N} \\ & \quad \mu_{nm} \geq 0, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{T} \end{aligned}$$

For link (n, \mathcal{D}_{nm}, d) , we define $c_{nmd}(\mathbf{p})$ as

$$c_{nmd}(\mathbf{p}) = p_{nm} \prod_{k: d \in \mathcal{N}_k, k \neq n} (1 - p_k) \quad (17)$$

where \mathbf{p} is the vector of random access probabilities, $\mathbf{p} = \{p_{nm}, (n, \mathcal{D}_{nm}) \in \mathcal{T}\}$. The formulation of \mathbf{P} for our guaranteed multicast problem is similar to the formulation for maximizing end-to-end proportional fairness in (unicast) random access networks, which is treated in [3] and [5]. Due to the form of $c_{nmd}(\mathbf{p})$, the problem is non-separable and non-convex in \mathbf{p} , which makes it difficult

to provide a distributed algorithm that converges to the global optimum. However, due to the similarity of this problem to the end-to-end proportional fairness problem, we can easily adapt the Dual-Based Algorithm provided in [3], which overcomes these difficulties and is shown to converge to the optimal p_{nm} .

We proceed by decomposing \mathbf{P} into two problems. The problem $\hat{\mathbf{P}}$ optimizes the same objective as \mathbf{P} but is parameterized by the set of (unicast) link throughputs $\boldsymbol{\mu}_{\mathcal{I}} = (\mu_{nmd}, (n, \mathcal{D}_{nm}, d) \in \mathcal{I})$, meaning essentially that the random access probabilities \mathbf{p} are assumed fixed. The problem $\hat{\mathbf{P}}$ is defined below.

$$\begin{aligned} \hat{\mathbf{P}} : \quad & \max \quad \sum_{(n, \mathcal{D}_{nm}) \in \mathcal{I}} w_{nm} \log \mu_{nm} \\ & \text{s.t.} \quad \mu_{nm} \leq \mu_{nmd}, \quad \forall d \in \mathcal{D}_{nm}, \forall (n, \mathcal{D}_{nm}) \in \mathcal{I} \\ & \quad \mu_{nm} \geq 0, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{I} \end{aligned}$$

Since \mathbf{p} is fixed, the problem $\hat{\mathbf{P}}$ is equivalent to a rate allocation problem in a wired network. Since the log function in the objective function is strictly concave and the constraints are linear in μ_{nm} , $\hat{\mathbf{P}}$ is a convex problem and has no duality gap. Then $\hat{\mathbf{P}}$ can be solved by the gradient projection method applied to its associated dual problem. This procedure is described below.

The solution to the problem $\hat{\mathbf{P}}$ will be a function of the link throughputs $\boldsymbol{\mu}_{\mathcal{I}}$. Let $\hat{U}(\boldsymbol{\mu}_{\mathcal{I}})$ denote the solution to $\hat{\mathbf{P}}$, defined as follows.

$$\hat{U}(\boldsymbol{\mu}_{\mathcal{I}}) = \max \left\{ \begin{aligned} & \sum_{(n, \mathcal{D}_{nm}) \in \mathcal{I}} w_{nm} \log \mu_{nm} : \mu_{nm} \leq \mu_{nmd}, \\ & \forall d \in \mathcal{D}_{nm}, \forall (n, \mathcal{D}_{nm}) \in \mathcal{I} \end{aligned} \right\} \quad (18)$$

The vector of link throughputs $\boldsymbol{\mu}_{\mathcal{I}}$ will ultimately be a function of \mathbf{p} . We define the function $\tilde{U}(\mathbf{p}) = \hat{U}(\mathbf{c}(\mathbf{p}))$, where $\mathbf{c}(\mathbf{p}) = (c_{nmd}(\mathbf{p}), (n, \mathcal{D}_{nm}, d) \in \mathcal{I})$ and the problem $\hat{\mathbf{P}}$ as follows.

$$\begin{aligned} \tilde{\mathbf{P}} : \quad & \max \quad \tilde{U}(\mathbf{p}) \\ & \text{s.t.} \quad 0 \leq p_{nm} \leq 1, \quad \forall (n, \mathcal{D}_{nm}) \in \mathcal{I} \\ & \quad p_n \leq 1, \quad \forall n \in \mathcal{N} \end{aligned}$$

Problem $\tilde{\mathbf{P}}$ is the problem we want to solve to determine \mathbf{p} . In the process of converging on the optimum \mathbf{p} , in each update we make we must optimize the vector of link throughputs $\boldsymbol{\mu}_{\mathcal{I}}$, which is accomplished by solving problem $\hat{\mathbf{P}}$. The algorithm will work at two different time scales. Let t_1 denote time instants in the larger time scale in which we update \mathbf{p} . Let t_2 denote time in the smaller

time scale, which is the time scale over which we solve $\hat{\mathbf{P}}$ through its dual problem.

We first describe the algorithm for solving the dual problem of $\hat{\mathbf{P}}$ when $\boldsymbol{\mu}_{\mathcal{I}} = \mathbf{c}(\mathbf{p}^{(t_1, t_2)})$. The Lagrangian is given by $L^{(t_1, t_2)}(\boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\lambda})$, where $\boldsymbol{\mu}_{\mathcal{I}} = (\mu_{nmd}, (n, \mathcal{D}_{nm}, d) \in \mathcal{I})$ denotes the vector of throughputs on the *multicast trees* in the network and $\boldsymbol{\lambda} = (\lambda_{nmd}, (n, \mathcal{D}_{nm}, d) \in \mathcal{I})$ denotes the vector of Lagrange multipliers, or link prices. We have

$$\begin{aligned} L^{(t_1, t_2)}(\boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\lambda}) = & \sum_{(n, \mathcal{D}_{nm}) \in \mathcal{I}} w_{nm} \log \mu_{nm} \\ & - \sum_{(n, \mathcal{D}_{nm}, d) \in \mathcal{I}} \lambda_{nmd} \left(\mu_{nm} - \mu_{nmd}^{(t_1, t_2)} \right). \end{aligned} \quad (19)$$

The solution to the dual problem of $\hat{\mathbf{P}}$ is

$$\boldsymbol{\lambda}^{*(t_1, t_2)} = \arg \min_{\boldsymbol{\lambda} \geq 0} \max_{\boldsymbol{\mu}_{\mathcal{I}}} L^{(t_1, t_2)}(\boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\lambda}) \quad (20)$$

Note that the Lagrangian can be rewritten as

$$\begin{aligned} L^{(t_1, t_2)}(\boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\lambda}) = & \sum_{(n, \mathcal{D}_{nm}) \in \mathcal{I}} \left(w_{nm} \log \mu_{nm} \right. \\ & \left. - \mu_{nm} \sum_{d \in \mathcal{D}_{nm}} \lambda_{nmd} \right) + \sum_{(n, \mathcal{D}_{nm}, d) \in \mathcal{I}} \lambda_{nmd} \mu_{nmd}^{(t_1, t_2)} \end{aligned} \quad (21)$$

and the first term is separable in μ_{nm} . The objective function for the dual problem of $\hat{\mathbf{P}}$ at $\boldsymbol{\mu}_{\mathcal{I}}$ is

$$\begin{aligned} D^{(t_1, t_2)}(\boldsymbol{\lambda}) = & \max_{\boldsymbol{\mu}_{\mathcal{I}}} L^{(t_1, t_2)}(\boldsymbol{\mu}_{\mathcal{I}}, \boldsymbol{\lambda}) \\ = & \sum_{(n, \mathcal{D}_{nm}) \in \mathcal{I}} \max_{\mu_{nm}} \left(w_{nm} \log \mu_{nm} - \mu_{nm} \sum_{d \in \mathcal{D}_{nm}} \lambda_{nmd} \right) \\ & + \sum_{(n, \mathcal{D}_{nm}, d) \in \mathcal{I}} \lambda_{nmd} \mu_{nmd}^{(t_1, t_2)} \end{aligned}$$

We can maximize $D^{(t_1, t_2)}(\boldsymbol{\lambda})$ over $\boldsymbol{\mu}_{\mathcal{I}}$ by setting

$$\mu_{nm} = \frac{w_{nm}}{\sum_{d \in \mathcal{D}_{nm}} \lambda_{nmd}} \quad (22)$$

The dual problem of $\hat{\mathbf{P}}$ can be solved using the gradient projection method, where λ_{nmd} are adjusted in the direction opposite to the gradient $\partial D^{(t_1, t_2)}(\boldsymbol{\lambda}) / \partial \lambda_{nmd}$. The update is performed as

$$\lambda_{nmd}^{(t_1, t_2+1)} = \left[\lambda_{nmd}^{(t_1, t_2)} - \gamma \frac{\partial D^{(t_1, t_2)}(\boldsymbol{\lambda}^{(t_1, t_2)})}{\partial \lambda_{nmd}} \right]^+ \quad (23)$$

where

$$\frac{\partial D^{(t_1, t_2)}(\boldsymbol{\lambda}^{(t_1, t_2)})}{\partial \lambda_{nmd}} = \frac{-w_{nm}}{\sum_{d \in \mathcal{D}_{nm}} \lambda_{nmd}^{(t_1, t_2)}} + \mu_{nmd}^{(t_1, t_2)} \quad (24)$$

and $\gamma > 0$ is the step size.

Once the gradient projection algorithm has converged to its solution $\lambda^{*(t_1, t_2)}$, we can update the p_{nm} as follows.

$$p_{nm}^{(t_1+1, t_2)} = p_{nm}^{(t_1, t_2)} + \alpha \sum_{(l, \mathcal{D}_{lk}, j) \in \mathcal{I}} \lambda_{lkj}^{*(t_1, t_2)} \frac{\partial c_{lkj}(\mathbf{p}^{(t_1, t_2)})}{\partial p_{nm}} \quad (25)$$

where

$$\frac{\partial c_{lkj}}{\partial p_{nm}} = \begin{cases} \prod_{\substack{i: j \in \mathcal{N}_i, \\ i \neq l}} (1-p_i), & (n, \mathcal{D}_{nm}, d) = (l, \mathcal{D}_{lk}, j) \\ -p_{lk} \prod_{\substack{i: j \in \mathcal{N}_i, \\ i \neq l, i \neq j}} (1-p_i), & j = n, j \in \mathcal{D}_{lk} \\ -p_{lk} \prod_{\substack{i: j \in \mathcal{N}_i, \\ i \neq l, i \neq n}} (1-p_i), & j \in \mathcal{N}_n \setminus \{n\}, j \in \mathcal{D}_{lk} \\ 0, & \text{else} \end{cases} \quad (26)$$

and α is the step size. Once \mathbf{p} has been updated, we again run the algorithm in (23) to converge on the optimal link prices, and repeat.

We summarize the proposed algorithm for guaranteed multicast throughput allocation in the following steps.

- (1) Pick arbitrary initial values for the access probabilities with $0 < p_{nm} < 1$.
- (2) Find λ^* using the algorithm described in (23) and (24).
- (3) Update the access probabilities according to (25) and (26).
- (4) Return to step (2) and repeat until the access probabilities p_{nm} have converged.

The convergence analysis of this algorithm follows directly from the proof of Theorem 1 in [3]. The analysis can be summarized as follows. Since the algorithm adjusts the access probabilities p_{nm} in the direction of the gradient, it yields a local optimal point for $\tilde{\mathbf{P}}$. This local optimal point for $\tilde{\mathbf{P}}$ is also a local optimal point for \mathbf{P} since the two problems are equivalent. The analysis in [3] shows that the local optimal point for \mathbf{P} is globally optimal.

V. AN EXAMPLE NETWORK

We now consider an example network to demonstrate the non-guaranteed and guaranteed approaches to optimal throughput allocation for random access multicast. The example network we consider is shown in Fig. 2, where the numbers correspond to the node identities. The lines in the figure indicate interference; for simplicity, we assume that interference is symmetric, i.e., $k \in \mathcal{N}_n \iff n \in \mathcal{N}_k$. The multicast flows we consider are shown in Table I, where nodes 3, 5 and 8 act as source

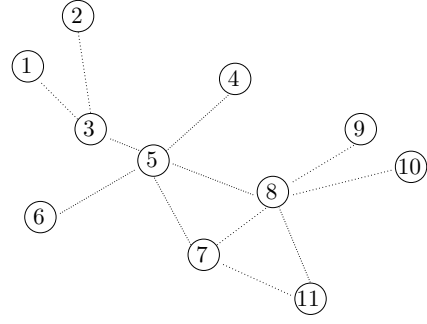


Fig. 2. An example network.

TABLE I
MULTICAST FLOWS FOR THE EXAMPLE NETWORK SHOWN IN FIGURE 2.

Source (n)	Tree (m)	Receivers
3	1	$\mathcal{D}_{31} = \{1, 2\}$
3	2	$\mathcal{D}_{32} = \{1, 2, 5\}$
5	1	$\mathcal{D}_{51} = \{3, 4\}$
5	2	$\mathcal{D}_{52} = \{6, 7, 8\}$
8	1	$\mathcal{D}_{81} = \{5, 7, 11\}$
8	2	$\mathcal{D}_{82} = \{9, 10\}$

nodes and two multicast flows emanate from each source. We assume that multicast traffic traversing the center of the network is given higher priority; thus in the examples we consider, we assign higher weights w_{nmd} and w_{nm} to multicast flows passing through node 5.

A. Non-guaranteed multicast

For non-guaranteed multicast, we assign a weight $w_{nmd} > 0$ to each link in the network and compute the optimal access probabilities according to (9). An example of the link weights, optimal access probabilities, and link throughputs μ_{nmd} as computed from (5) are shown in Table II. Due to the large weights w_{nmd} associated with the multicast flows emanating from node 5, node 5 is assigned a large access probability with $p_5 > 0.9$. As a result, the value of the link throughput μ_{nmd} is smallest for links $(3, 2, 5)$, $(8, 1, 5)$, and $(8, 1, 7)$, where node 5 is either a receiver or causes interference. The value of the link throughput μ_{nmd} is largest for links on the edge of the network, which do not suffer interference.

B. Guaranteed multicast

For guaranteed multicast, we assign a weight $w_{nm} > 0$ to each multicast tree and make use of the proposed

TABLE II

OPTIMAL ACCESS PROBABILITIES AND LINK THROUGHPUT VALUES FOR NON-GUARANTEED MULTICAST IN THE EXAMPLE NETWORK. A " INDICATES THAT AN ENTRY IS THE SAME AS THE ENTRY IN THE ROW ABOVE IT.

Link	w_{nmd}	p_{nm}	μ_{nmd}
(3, 1, 1)	0.5	0.25	0.25
(3, 1, 2)	0.5	"	0.25
(3, 2, 1)	0.5	0.5	0.5
(3, 2, 2)	0.5	"	0.5
(3, 2, 5)	1.0	"	0.0154
(5, 1, 3)	2.0	0.4615	0.1154
(5, 1, 4)	1.0	"	0.4615
(5, 2, 6)	0.5	0.4615	0.4615
(5, 2, 7)	1.0	"	0.1846
(5, 2, 8)	1.5	"	0.1846
(8, 1, 5)	1.0	0.4	0.0077
(8, 1, 7)	0.5	"	0.0308
(8, 1, 11)	0.5	"	0.4
(8, 2, 9)	0.5	0.2	0.2
(8, 2, 10)	0.5	"	0.2

algorithm to compute the random access probabilities. The link weights that we considered, along with the values of p_{nm} and μ_{nmd}^{min} that the algorithm reached at convergence, are shown in Table III.

In implementing the algorithm, we set the step sizes at $\alpha = 5 \times 10^{-4}$ and $\gamma = 25$. We have assumed that the gradient projection algorithm has converged to its solution $\lambda^{*(t_1, t_2)}$ once the variation in all λ_{nmd} values is less than 5×10^{-3} . With these parameters, we observed that the convergence of the gradient projection algorithm required at most 180 iterations, and typically required fewer than 20 iterations. As shown in Figs. 3 and 4, the algorithm to update p_{nm} converges on the optimal p_{nm} and μ_{nmd}^{min} for guaranteed multicast. In those two figures, the number of iterations (along the x-axis) refers to the number of updates to the values of p_{nm} , and we observe that at most 300 iterations are needed to reach convergence. For implementation in practical scenarios, the convergence speed can be adjusted by varying the step sizes, particularly by choosing the step size γ individually for each link (n, \mathcal{D}_{nm}, d) .

Once again, due to the large weights assigned to the multicast flows emanating from source 5, it is assigned a large access probability with $p_5 \approx 0.65$ and the guaranteed multicast throughput is largest for trees $(5, \mathcal{D}_{51})$

TABLE III

OPTIMAL ACCESS PROBABILITIES AND MINIMUM LINK THROUGHPUT VALUES FOR GUARANTEED MULTICAST IN THE EXAMPLE NETWORK.

Tree	w_{nm}	p_{nm}	μ_{nmd}^{min}
$(3, \mathcal{D}_{31})$	1	0.0952	0.0952
$(3, \mathcal{D}_{32})$	2	0.3178	0.0718
$(5, \mathcal{D}_{51})$	3	0.2040	0.1198
$(5, \mathcal{D}_{52})$	3	0.4549	0.3012
$(8, \mathcal{D}_{81})$	2	0.2330	0.0466
$(8, \mathcal{D}_{82})$	1	0.1048	0.1048

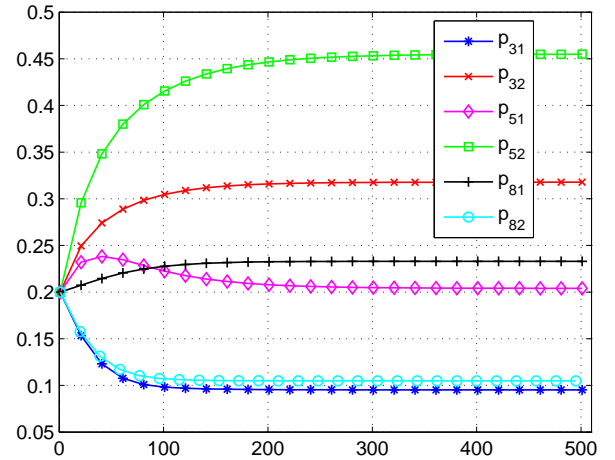


Fig. 3. Values of p_{nm} versus iteration number for the proposed algorithm to compute random access probabilities for guaranteed multicast.

and $(5, \mathcal{D}_{52})$. The multicast flows on the edge of the network obtain a moderate throughput due to the lack of interference. The guaranteed throughput is smallest for flows $(3, \mathcal{D}_{32})$ and $(8, \mathcal{D}_{81})$ due to the moderate weight assigned to those flows coupled with the interference from node 5.

VI. CONCLUSION

In this work we propose schemes to assign the random access probabilities to nodes in a wireless network in order to optimally allocate the *multicast* throughput. For applications where non-guaranteed multicast is sufficient, the optimal throughput allocation can be computed in a simple, distributed manner similar to the manner in which throughput is allocated for *unicast* transmission. If guaranteed multicast transmission is required, the throughput-optimal strategy involves fountain coding of

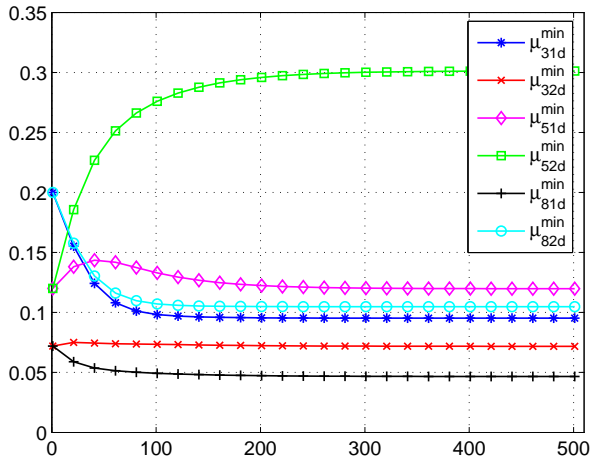


Fig. 4. Values of μ_{nmd}^{\min} versus iteration number for the proposed algorithm to compute random access probabilities for guaranteed multicast.

packets and a distributed algorithm that converges to the optimal access probabilities. We have assumed that fountain coding is performed with $K \rightarrow \infty$, which would yield an infinite delay. An important follow-up to this work would be to analyze the guaranteed throughput performance for fountain coding with finite K , and thus finite delay. Furthermore, it would be useful to extend our work to end-to-end multicast flows. Finally, future work in this area could involve a performance evaluation of guaranteed multicast for unslotted, carrier-sensing-based random access.

ACKNOWLEDGMENT

The authors would like to thank Anthony Ephremides for useful discussions about this work.

REFERENCES

- [1] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportionally fair rates using local information in ALOHA networks," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1858-1862, 2004.
- [2] P. Gupta and A.L. Stolyar, "Optimal throughput allocation in general random-access networks," *Proc. Conference on Information Sciences and Systems (CISS)*, Princeton, March, 2006.
- [3] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," *Proc. MobiHoc*, Urbana-Champaign, May, 2005.
- [4] J. Mo and J. Walrand, "Fair end-to-end window-based contention control," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 556-567, October, 2000.
- [5] J.-W. Lee, M. Chiang, and A. R. Calderbank, "Jointly optimal congestion and contention control based on network utility maximization," *IEEE Communications Letters*, Vol. 10, no. 3, pp. 216-218, March 2006.
- [6] P. Chaporkar and S. Sarkar, "Wireless multicast: theory and approaches," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1954-1972, June 2005.
- [7] P. Chaporkar and S. Sarkar, "Minimizing delay in loss-tolerant MAC layer multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4701-4713, October 2006.
- [8] K. Kar, S. Sarkar, and L. Tassiulas, "Optimization based rate control for multirate multicast sessions," *Proc. INFOCOM*, Alaska, 2001.
- [9] M. Luby, "LT-codes," *Proceedings of the ACM Symposium on Foundations of Computer Science (FOCS)*, 2002.
- [10] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551-2567, 2006.