The Multicast Capacity of Large Multihop Wireless Networks

Srinivas Shakkottai
Dept. of ECE, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
sshakkot@uiuc.edu

Xin Liu
Dept. of Computer Science
University of California at Davis
liu@cs.ucdavis.edu

R. Srikant
Dept. of ECE, and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
rsrikant@uiuc.edu

ABSTRACT

We consider wireless ad hoc networks with a large number of users. Subsets of users might be interested in identical information, and so we have a regime in which several multicast sessions may coexist. We first calculate an upper-bound on the achievable transmission rate per multicast flow as a function of the number of multicast sources in such a network. We then propose a simple comb-based architecture for multicast routing which achieves the upper bound in an order sense under certain constraints. Compared to the approach of constructing a Steiner tree to decide multicast paths, our construction achieves the same order-optimal results while requiring little location information and no computational overhead.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design—wireless communication

General Terms
Theory and Algorithms

Keywords
Ad hoc networks, multicast flows, Steiner tree, asymptotic capacity

1. INTRODUCTION

Wireless ad hoc networks are of use when there is a lack of fixed communication infrastructure. Such situations might arise in calamity environments, sensor network applications, and a variety of other civilian and military contexts. Most of the research in large ad hoc networks has focused on unicast data transfers, either between users, or between a user and fixed infrastructure. One area that has received little attention is the use of such ad hoc networks for multicast data transmission. In many applications, multicast data transfer is more predominant than unicast data transfer. In military networks it is often stated that multicast traffic dominates due to the need for group communications. In the civilian context, an emerging application that has already been tested in many universities is the use of wireless ad hoc networks to broadcast replays during football games. A situation like a football game would have a large number of spectators, each having a mobile device and a desire for a replay of an important moment in the game. There is almost no infrastructure available from which they could obtain such data, and there is a strong incentive to form an ad hoc network for this purpose. Some of the users might be close to data sources (perhaps if they were close to an Internet access point), and they would act as sources for the multicast traffic. Other nodes would act as relays and sinks for the data. The questions arise as to how many multicast sessions can be supported by such a network, what the total capacity of the network would be, and how to achieve the capacity in a simple and practical manner.

Consider Figure 1. There is a finite area with a number of wireless nodes. There are 3 multicast flows in progress, with nodes receiving each multicast flow labeled 1, 2 and 3, respectively. The sources of these flows are labeled as $S_1$, $S_2$, and $S_3$, respectively. The hops are labeled with the sessions that they carry. Some nodes may be neither sources nor destinations and merely act as relays (unlabeled nodes). Suppose each hop, if scheduled, could carry one bit per time slot. The first constraint on the system capacity is that we may not be able to schedule all the hops simultaneously due to interference. Even if all hops could be scheduled simultaneously, we see that the throughput of each source is at most 0.5 bits per time slot in this example, since all the multicast flows contain at least one node that is shared with one other multicast flow. The total throughput of the system would then be 1.5 bits per second. Thus, we see that there are two main sources of interference that limit the multicast capacity of the network:

1. The channel interference constraint: while we must have a sufficiently large transmission radius for the network to be connected, a larger radius means a transmitting node interferes with more nodes, which lim-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MobiHoc '07, September 9–14, 2007, Montréal, Québec, Canada.
Copyright 2007 ACM 978-1-59593-684-4/07/0009 ...$5.00.
as a function of the number of multicast sources. To the best of our knowledge, the only prior work that deals with scaling laws in wireless networks with multicast flows is [14]. However, they use a Steiner tree approach to construct a multicast tree, and their results only apply to a single multicast flow in a large network. Our main contribution is to propose a simple architecture that achieves the same capacity in the order sense. In addition, we also consider the scaling law as a function of source nodes.

Main Results

We first derive an upper bound on the multicast capacity of a wireless ad hoc network along the lines of [14]. In the Gupta-Kumar model, the number of source-destination pairs is \( n \). To compare and contrast our result with theirs, we too assume that the number of S-D pairs in our network with multicast flows is \( n \). In particular, we assume that the number of multicast sources is \( n \) for some \( \epsilon > 0 \), and the number of receivers per multicast flow is \( n^{1-\epsilon} \). We express our results in the order sense. We say \( f(n) = O(g(n)) \) with high probability (w.h.p) if given \( \delta > 0 \), \( \exists c, \) and \( (m(\delta)) \) such that

\[
P\{ f(n) \leq c \cdot g(n) \} \geq 1 - \delta \quad \forall n \geq m(\delta).
\]

Similarly, we say \( f(n) = \Theta(g(n)) \) with high probability (w.h.p) if given \( \delta > 0 \), \( \exists c_1, \) and \( m(\delta) \) such that

\[
P\{ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \} \geq 1 - \delta \quad \forall n \geq m(\delta).
\]

Under the above model we show that the upper bound on the sum of the source rates that the network can support is

\[O\left(\frac{\sqrt{n}}{\sqrt{\log n}}\right)\] w.h.p.,

with a per flow throughput capacity of

\[O\left(\frac{1}{\sqrt{n \log n}}\right)\] w.h.p.

Note that this is a direct generalization of the Gupta-Kumar upper bound for the per flow throughput capacity for unicast, which is

\[O\left(\frac{1}{\sqrt{n \log n}}\right)\] w.h.p.

Our upper bound is a simple extension of the arguments in [1, 14]. The achievability of the upper bound is the main contribution of this paper. To achieve the upper bound, we propose a simple routing architecture to transfer multicast data in the network. The architecture consists of a tree called the multicast comb, which is constructed independent of the receiver locations. The receivers then complete the multicast tree by attaching themselves to the comb using shortest path routing in a small vicinity. Using this simple architecture, we show that the achievable throughput matches the upper bound in an order sense if each flow is allowed to drop an arbitrarily small fraction of its receivers or we can pose constraints on the locations of the source and destination nodes.

Organization of the Paper

We begin in Section 2 with a description of the system model that we consider. We derive the upper bound on multicast
capacity and compare it to the unicast upper bound in Section 3. We then proceed in Section 4 to design a simple architecture that achieves the upper bound in the order sense if the sources and destinations can be chosen suitably. However, since our goal is to study the case of randomly chosen sources and destinations, in Section 5 we find the lower bound on the achievable transmission rate. We conclude with possible extensions in Section 6.

2. SYSTEM MODEL

We consider a square of unit area, in which $n$ wireless nodes are located randomly. There are $n_s$ multicast sources chosen randomly from the $n$ nodes, and each of these sources is associated with $n_d$ destination nodes also chosen randomly. Thus, the total number source-destination pairs is $n_s n_d$. We assume that time is slotted, and the capacity of the wireless channel is such that a hop, if scheduled, can successfully transmit 1 bit per slot. We use the protocol model to model interference between transmissions, as proposed in [1]. The model is illustrated in Figure 2. We assume that all nodes choose identical transmit radii $r$, which is large enough for the network to be connected. Suppose that node $i$ transmits to node $j$. Node $j$ receives the transmission successfully if every other node that transmits simultaneously is at a distance of at least $(1 + \Delta)r$ from $j$. This implies that circles of radius $\frac{\Delta}{2}$ around each receiver must be disjoint [1].

![Figure 2: Illustration of the protocol model for wireless transmission. The assumption is that a transmission would be successful as long as there is no other transmission in a circle of radius $(1 + \Delta)r$ from the intended receiver.](image)

When $n_s = n - 1$ and $n_d = 1$ the model is the same as a unicast problem with $\Theta(n)$ sources. When $n_s = 1$ and $n_d = n - 1$ the source can broadcast its data in one hop to all the destinations with a transmission radius of $\sqrt{2}$. We would like to understand the capacity scaling law and its practical achievability between these two extremes.

3. UPPER BOUND ON MULTICAST CAPACITY IN RANDOM NETWORKS

Our first objective is to derive an upper bound on the throughput capacity of multicast wireless networks. The derivation is similar to that in [1, 14], with a small modification to account for the fact that we have multiple multicast flows. Consider a bit $b$ originating at source $i$. Let the number of hops required by $b$ to reach all its final destinations be denoted $H_i(b)$. Then the total number of hops used by all bits to reach their respective destinations is $\sum_i H_i(b)$. Note that this could take several time slots. We would like to know if these bits can be transmitted in some time interval $T$. Now, since our assumption is that each transmission can support the transfer of one bit, the number of transmissions required for this to happen is the same as the number of hops required. Let the total number of simultaneous transmissions possible in the system be $S$. Note that $S$ is independent of time as we assume that system does not change with time. From the above argument, we require that for the bits to reach their destinations in an interval $T$,

$$\sum_{i,b} H_i(b) \leq TS. \quad (1)$$

Thus, if the rate at which sources can generate bits is $\lambda$ (common to all sources), the number of bits generated by these sources in time interval $T$ is simply $\lambda T n_s$. If we can show that the total number of hops required to support these bits in time interval $T$ is at least $\lambda T n_s H(b)$ with high probability for some $H(b)$, then our necessary condition is now

$$\lambda T n_s H(b) \leq TS. \quad (2)$$

Now, if we take the radius of transmission to be $r$, since our square is of size 1, the maximum number of simultaneous transmissions can be bounded as

$$S \leq \frac{4}{\pi \Delta^2 r^2}. \quad (3)$$

Hence, from the above and (2) the source rate in of a random ad hoc network in both unicast and multicast cases is upper bounded as

$$\lambda \leq \frac{4}{\pi \Delta^2 r^2 n_s H(b)}. \quad (4)$$

The questions that need to be answered to obtain a useful bound on the throughput capacity are: (i) what should the radius $r$ be?, and (ii) what is the number of hops required to reach all destinations?

It has been shown in [1] that the radius $r$ has to be chosen such that

$$r = \sqrt{\frac{\log n + \kappa_n}{n}}$$

(where $\limsup_n \kappa_n < +\infty$) to guarantee connectivity of nodes with high probability.

We now need the following results from [15, 16] for the length of the multicast tree in random networks.

Result 1 Suppose we drop $m$ nodes in a unit square, where the position of each node is chosen uniformly at random in the square. Then as $m \to \infty$ the length of the minimum spanning tree $L(m)$ satisfies

1. $\mathbb{E}(L(m)) \sim C_1 \sqrt{m}$ (Lemma 3.3 from [16]), and
2. \( \text{Var}(L(m)) \leq C_2 \log m \) (Lemma 4.1 from [16]),

where \( C_1 \) and \( C_2 \) are positive constants, and the notation \( \sim \) is used to denote asymptotic equality. Also, the length of the optimal Euclidean Steiner tree connecting all the \( m \) nodes \( S(m) \geq \frac{C}{2} L(m) \) [15].

**Theorem 1.** The throughput of each multicast source in a random wireless ad hoc network is upper bounded by

\[
O \left( \min \left( 1, \frac{\sqrt{n}}{n_s \sqrt{n_d \log n}} \right) \right) \text{ w.h.p.}
\]

as \( n \to \infty \), where \( n_s \) and \( n_d \) are integer functions of \( n \) satisfying \( n_s n_d \leq n \),

\[
\lim_{n \to \infty} \frac{\log n_d}{n_s} = 0,
\]

and \( n_s, n_d \to \infty \) as \( n \to \infty \).

**Proof.** Consider (4). We need the radius of transmission \( r \) and the total number of source destination pairs to be the same in both cases. This would enable us to characterize the gain that could potentially be achieved using multicast. Since the unicast regime consists of \( n \) source destination pairs, we take \( n_s n_d = n \). In particular, we take \( n_d = n^{1-\epsilon} \) and \( n_s = n^{\epsilon} \), where \( 0 \leq \epsilon \leq 1 \). We have the following corollary:

**Corollary 2.** The throughput of each multicast source in a random wireless ad hoc network is upper bounded by

\[
\Theta \left( \frac{1}{\sqrt{n \log n}} \right) \text{ w.h.p.}
\]

Notice that as \( \epsilon \to 1 \), the multicast throughput capacity is essentially the same as the unicast capacity as it should be. Also notice that as \( \epsilon \to 0 \), the point of using multicast is lost since all nodes could be reached by one broadcast hop. As an example, consider the case where \( n = 10,000 \) and \( \epsilon = 1/2 \) (i.e., there are 100 sources, each with 100 destinations). The per-source throughput that can be achieved by using multicast is 10 times that of unicast.

We now move to the problem of designing a simple scheduling and routing algorithm that could be used to implement the multicast idea. We first show that in a network where we are allowed to choose the sources and destination, the multicast capacity can easily be attained. We then consider the case of randomly chosen sources and destinations.

**4. ORDER OPTIMAL COMB STRUCTURE**

We develop a simple architecture, whereby we may achieve the upper bound found above (in the order sense), when we are allowed to select the source and destination nodes appropriately. As before, we study the system with \( n_s \) multicast sources each with \( n_d \) destinations, and then consider the case when \( n_s = n^\epsilon \) and \( n_d = n^{1-\epsilon} \). The main features of our construction are as follows.

- We first divide the region into squarelets in the manner of [2]. The squarelet size is large enough so that there is at least one node in each squarelet with high probability, and nodes in adjacent squarelets are capable of communicating with each other. Let the length of a squarelet be \( s \).

- The scheduling algorithm will be chosen such that the squarelets that are \( K s \) apart are scheduled simultaneously, where \( K \) is chosen such that the wireless interference constraint is satisfied.

- We construct multicast comb structures using the constructed squarelets. There is one comb corresponding
to each multicast flow. The combs are constructed so that at most 2 flows pass through a squarelet, so as to keep interference between flows minimal. In the comb structure, the width between two consecutive comb teeth is determined by $n_d$, the number of destinations per multicast source. Each multicast destination node can reach any one tooth on the comb to receive multicast data. This can be done in the vicinity of the destination node using any routing algorithms, such as shortest path routing. To construct the comb, the location of the destination nodes is not required and no central control is needed. In addition, the cost of the comb structure is the same as an optimal Steiner tree in an order sense.

We now present the details of the architecture and derive its throughput capacity. Recall that the length of a squarelet is $s_n$. We have the following useful result from [2]:

**Result 2** For a squarelet size

$$s_n = \frac{\sqrt{3} \log n}{\sqrt{n}},$$

no squarelet is empty with probability at least $1 - \frac{1}{n^2}$.

We need to ensure that a node in one corner of a squarelet can transmit to a node in the opposite corner of an adjacent squarelet, i.e., the transmission radius is chosen as $r = \sqrt{5}s_n$. We can guarantee successful reception at a receiving node, if no other transmission takes place within a distance of $\sqrt{5}s_n(1 + \Delta)$. Recall that $Ks_n$ is the distance such that squarelets that are this distance apart can be scheduled simultaneously. Then, as illustrated in Figure 3, we have

$$(K - 2)s_n \geq (1 + \Delta)\sqrt{5}s_n \Rightarrow K = 2 + (1 + \Delta)\sqrt{5}.$$

We call the subset of squarelets a distance of $K$ squarelets from each other, capable of simultaneous transmission as an equivalence class. So the number of such equivalence classes is $K^2$. We then have a system in which the periodicity with which any squarelet is scheduled is $K^2$ time slots.

Using the above idea, we construct the following comb structure for multicast traffic. Suppose that we are able to select the source and receiver nodes of the multicast. We first construct multiple combs, one for each multicast flow as shown in Figure 4. In the figure we have illustrated two multicast combs (one lightly shaded (cyan), and the other dark (magenta)) corresponding to two multicast flows. The distance between the teeth of each comb is chosen to be $\frac{1}{\sqrt{n_d}}$. The Euclidean length of comb routing for a particular multicast comb is

$$\sqrt{n_d} + 1 = \Theta(\sqrt{n_d}). \quad (5)$$

From Result 1, we have that the Euclidean length of the optimal Steiner tree is $\Theta(\sqrt{n_d})$, which is the same as our comb structure in an order sense. The construction of the Steiner tree requires global location information and a centralized controller. In comparison, the construction of the comb structure only requires information on the number of destination nodes and does not require a central controller to compute the tree.

Each comb is positioned one squarelet farther to the right and below to the previous one. Note that by using this structure, we have ensured that the maximum number of multicast flows that use any particular squarelet is 2. Since each squarelet is scheduled with a periodicity of $K^2$ slots and the channel has a capacity of 1 bit per time slot (if scheduled), if only 2 multicast flows share a squarelet, their individual throughputs would be $1/2K^2$ bits per time slot.

Since we can choose the locations of the source and destination nodes as desired, we let the source and destinations of a particular multicast flow to lie on its corresponding comb. For example, in Figure 4, the sources and destinations associated with the lightly shaded (cyan) comb would lie somewhere on the comb. As the distance between the teeth of a comb is $\sqrt{n_d}$, the number of such combs that can be constructed is $\frac{1}{\sqrt{n_d}}$. Choosing $n_d = n^{1-\epsilon}$ and $s_n = \frac{\sqrt{3} \log n}{\sqrt{n}}$, the proposed system can accommodate a total of

$$\frac{\sqrt{n^2}}{\sqrt{3} \log n} \quad (6)$$

sources without using one comb for multiple flows. As explained above, since there are a maximum of 2 flows using each squarelet, the sources can each transmit at a rate of $1/2K^2$. This results in a total source rate of

$$\frac{\sqrt{n^2}}{2K^2 \sqrt{3} \log n}, \quad (7)$$

which is of the same order of magnitude as the upper bound calculated in the previous section.

If $n_s > \sqrt{n^2}/\sqrt{3 \log n}$, then each comb is reused over multiple multicast sources. For example, if the number of sources were set to be $n_s = n^\epsilon$ as in the previous section, each comb would have to carry the traffic of $\sqrt{3n^\epsilon \log n}$ sources, giving each source a throughput of

$$\frac{1}{2K^2 \sqrt{3n^\epsilon \log n}}.$$
Figure 4: The comb idea for multicast data transfer. Each comb carries a data from a different source. The size of the squarelet places a fundamental limit on the number of possible coexistent combs.

which is of the same order as the per flow throughput of the previous section (See Corollary 2).

Notice that in this scheme, the throughput of the system increases with the number of multicast sources at a constant rate until \( n_s = \sqrt{\frac{n}{\log n}} \). After this point, the aggregated source rate remains constant as shown in Figure 5, for a given value of \( n_d \).

We have proposed a simple multicast architecture, and shown that its throughput in ideal circumstances is identical (in the order sense) to the upper bound. However, we have yet to study its performance when we are not at liberty to place the sources and destinations. We proceed to answer this question in the following section.

5. ACHIEVABLE MULTICAST CAPACITY IN RANDOM NETWORKS

We have just seen how the comb architecture is capacity-achieving in the case where source-destination placements can be made as desired. We will now study the case where the sources and destinations are randomly chosen. In this case the both sources and destinations must reach the comb in a multi-hop manner. Once they reach the comb, they would have access to the multicast traffic on that comb. The limiting factor is that as the number of such access paths increase, the intersections between them does as well, leading to reduction of throughput for the intersecting flows. In this section, we will study this effect. We will first show that there are squarelets in which the number of such intersections is large. We will then show that the number of such squarelets is a small percentage of the whole, and we can drop nodes in these squarelets to achieve order optimal throughput for the remaining nodes.

Recall that in the previous section we showed that the capacity achieving number of sources was \( n_s = \sqrt{\frac{n}{\log n}} \) with each source associated with \( n_d = \frac{n}{\sqrt{\log n}} \) destinations. Also recall that if the number of sources is greater than this value, we have comb reuse with the total capacity remaining constant. We will study the system with the capacity achieving parameters above, although our derivations are valid for a general \( n_s \) and \( n_d \). We index the co-existing combs of Figure 4 from 1 to \( \sqrt{\frac{n}{\log n}} \). We associate each source with one of the combs, and both the source and its destinations choose the shortest path to reach one of the teeth of the chosen comb. As shown in Figure 6, these paths are simple to construct (they are either above or below the teeth). Note that the maximum length of the path is \( \frac{1}{\sqrt{2}} \sqrt{\frac{n}{\log n}} \), which means that the increase in length of the multicast tree due to these branch paths would be just \( n_d \times \frac{1}{\sqrt{2}} \sqrt{\frac{n}{\log n}} = \Theta(\sqrt{n d}) \). Thus, comparing with (5), we see that Euclidean length of the tree is unaffected in the order sense by the branch paths.

As explained in the example in the introduction, the transmission rate of the sources (such that all their destinations can receive all the bits that they transmit) depends on the number of multicast flows that share the same squarelets. To find the achievable region, we need to know the maximum number of different multicast flows that share the same squarelet. We can then divide the throughput capacity by
this number to find the achievable throughput. We first need a standard result on the so-called occupancy problem (See [17] for example):

**Result 3** Suppose that we have \( m \) balls, and we drop them uniformly at random into \( m \) bins. Then the maximum number of balls in any bin is upper bounded as

\[
O \left( \frac{\log m}{\log \log m} \right),
\]

with probability \( 1 - 1/m \to 1 \) as \( m \to \infty \).

We are now ready to find the maximum number of multicast flows sharing each squarelet. We have the following theorem:

**Theorem 3.** Given that the network is connected, the maximum number of other interfering multicast flows for any particular multicast flow is

\[
\frac{2n}{\sqrt{3n^3 \log n}}
\]

with probability \( 1 - \frac{\sqrt{3n^3 \log n}}{2n} \to 1 \) as \( n \to \infty \).

**Proof.** We already know that the comb structure itself requires that 2 multicast flows share some squarelets. Given a particular multicast flow, we would like to know how many others share squarelets with it. Consider Figure 7. Since both sources and destinations of a particular multicast flow would use the shortest vertical path to connect to the relevant comb, we can divide the area into “bins” associated with a particular flow. The size of the bin is half the distance between the teeth of the comb, and is hence \( 1/(2\sqrt{n_d}) \). We have indicated a few of the bins using arrows in the Figure 7. The number of interfering multicast flows is the number of nodes belonging to different multicast flows that fall into each bin.

![Figure 7: The number of available bins for dropping nodes when considering a particular source. The arrows show the extent of a few bins associated with the comb. The number of nodes associated with different multicast flows in each bin should not be too large if the throughput capacity is to be near optimal.](image)

We first find the number of bins present. Clearly, the number of strips (areas between the dark horizontal lines and the nearest tooth of the comb) in Figure 7 is \( 2\sqrt{n_d} \). Then the number of bins associated with each multicast flow is just \( 2\sqrt{n_d}/s_n \), which simplifies to

\[
\frac{2n}{\sqrt{3n^3 \log n}}.
\]

We now drop nodes into these bins at random. Although we really want to know the number of types of nodes in each bin (each type corresponding to a different multicast flow), we can find an upper bound by assuming that each node belongs to a different multicast flow. Thus, the number of node types is upper bounded by

\[
(n_s - 1)(1 + n_d) \leq 2n_s n_d,
\]

since \( n_d \geq 1 \). When \( n_s \) is taken to be capacity achieving value (see (6)) of

\[
\frac{\sqrt{n^3}}{\sqrt{3\log n}}
\]

and \( n_d \) taken as \( n^{1-\epsilon} \), the number of node types that are dropped is upper bounded by

\[
\frac{2n}{\sqrt{3n^3 \log n}},
\]

which is identical to the number of bins given in (8). Thus, we can look at the system as an occupancy problem with equal numbers of balls and bins. The result follows from Result 3.

We have just seen that multicast flows might have a large number of other interfering flows, causing a loss of throughput. But how many squarelets would actually have such overcrowding? If the fraction of overcrowded squarelets were small, then we could simply drop the destinations belonging to these squarelets. Thus, we propose to use a majority rule, which can provide a high multicast rate (on the same order of the capacity upper bound) to a majority of users if we are allowed to sacrifice an arbitrarily small percentage of users in overcrowded areas. Let the probability that nodes in any particular squarelet do not receive the order optimal throughput be \( P_{th} \). We show below that we can achieve the goal of order optimal throughput for the remaining nodes for any \( P_{th} > 0 \).

**Theorem 4.** Given any threshold \( P_{th} > 0 \), the probability that nodes in a squarelet receive the order optimal per flow throughput capacity of

\[
O \left( \frac{1}{n^3 \log n} \right)
\]

is at least \( 1 - P_{th} \).

**Proof.** Consider Figure 7 again. Consider any squarelet that is on a multicast tree. There would be a number of branches passing through this squarelet, where each branch is a path taken by some node to reach its desired comb. The length of any branch is at most \( 1/2\sqrt{n_d} \), which is the same as \( \sqrt{n}/2\sqrt{3\log n} \) squarelets. Call this branch length in squarelets as \( l \). Let \( X \) be the number of random branches passing through the squarelet. The value of \( X \) decides the crowdedness. Now, the probability that any particular node
lands in any particular squarelet is the number of nodes divided by the number of squarelets, and is given by \( p = (3n_l \log n)/n \). Let \( X_{ij} \) be a random variable, where
\[
X_{ij} = \begin{cases} 
1 & \text{with probability } p \\
0 & \text{with probability } 1 - p
\end{cases}
\]
which is the probability that squarelet \( j \) is on multicast stream \( i \). We then have
\[
X = 1 + \sum_{i=1}^{n_s - 1} \sum_{j=1}^{l} X_{ij} \\
\leq 1 + \sum_{i=1}^{n_s - 1} X_{ij} \\
\leq X',
\]
where we have used the fact that the length of a branch is at most \( l \) squarelets. Let \( \rho \) be the average load on a node if destinations are evenly distributed. We have \( \rho = n_s \times l \times p \). Substituting \( n_s = n^s \), \( n_l = n^{1-s} \), and the values of \( l \) and \( p \) as calculated above, we get \( \rho = \sqrt{3n^s \log n}/2 \). Given some \( W \in \mathbb{R}^+ \) and using the Chernoff bound, we have
\[
P(X \geq W\rho) \leq \frac{\mathbb{E}(e^X)}{e^{W \rho}}.
\]
We calculate \( \mathbb{E}(e^X) \) next as follows:
\[
\mathbb{E}(e^X) = \mathbb{E}(e^{X'}) \\
= \sum_{i=0}^{n_s - 1} \binom{k}{n_s l} (1 - p)^{n_s l - k} e^k \\
= (1 + (e - 1)p)^{n_s l}.
\]
Therefore,
\[
P(X \geq W \max(\rho, 1)) \leq \frac{(1 + (e - 1)p)^{n_s l}}{e^{W \max(\rho, 1)}} \tag{1} \\
\leq 2^{e^{n_s l} (e - 1)p} \\
= 2^{e^W \max(p, 1)} \\
\leq e^{(e - 1 - W) \max(p, 1)} \tag{2} \\
\leq P_{th},
\]
where (1) holds for large enough \( n_s \). Therefore, for a given \( P_{th} \), we can find a finite \( W \) large enough so that (2) holds. In other words, with probability \( 1 - P_{th} \), we can guarantee that a node can achieve a multicast rate of \( 1/(W \max(\rho, 1)) \), where
\[
W \geq e - 1 - \frac{\ln P_{th}}{\max(\rho, 1)}.
\]
Since \( \rho = \sqrt{3n^s \log n}/2 \), this value is just
\[
\frac{2}{W \sqrt{3n^s \log n}}.
\]
Thus, the achievable rate is the same as the upper bound within a constant factor \( 2/W = \Theta((\ln 1/P_{th})^{-1}) \).

We have thus shown that by dropping an arbitrarily small fraction of the nodes, we can achieve order optimal throughput for the rest of the nodes. Note that as \( P_{th} \) approaches 0, the per-source throughput also approaches zero. However, the rate of decrease is slow: it is proportional to the logarithm of \( 1/P_{th} \). This kind of policy would be acceptable in real situations where it is important that most of the users obtain high rate transmission, rather than having to cut down the rate for all users so as to satisfy a small percentage of overcrowded areas.

6. CONCLUSIONS

In this paper we have developed an analytical framework for studying the multicast capacity of wireless ad hoc networks. We started with a comparison of the unicast case that has been studied in detail earlier, and showed how the multicast capacity is a function of the number of multicast sources and destinations. We developed a new and simple scheme that we called the comb architecture that would achieve this upper bound if we were at liberty to place the sources and destinations. We also studied the random network case and showed that the price paid in terms of throughput capacity for the simple and robust architecture is not high in the order sense. In the future we would like to study experimental wireless multicast networks.

7. REFERENCES


