Multicast Capacity of Wireless Ad Hoc Networks Under Gaussian Channel Model

Xiang-Yang Li, *Senior Member, IEEE*, Yunhao Liu, *Senior Member, IEEE*, Shi Li, and ShaoJie Tang

***Abstract*—We study the multicast capacity of large-scale random extended multihop wireless networks, where a number of wireless nodes are randomly located in a square region with side length $a = \sqrt{n}$, by use of Poisson distribution with density 1. All nodes transmit at a constant power P, and the power decays with attenuation exponent $\alpha > 2$. The data rate of a transmission is determined by the SINR as $B \log(1 + \mathrm{SINR})$, where B is the bandwidth. There are n_s randomly and independently chosen multicast sessions. Each multicast session has k randomly chosen terminals. We show that when $k \leq \theta_1 \frac{n}{(\log n)^{2\alpha+6}}$ and $n_s \geq \theta_2 n^{1/2+\beta}$, the capacity that each multicast session can achieve, with high probability, is at least $c_8 \frac{\sqrt{n}}{n_s\sqrt{k}}$, where θ_1, θ_2, and c_8 are some special constants and $\beta > 0$ is any positive real number. We also show that for $k = O(\frac{n}{\log^2 n})$, the per-flow multicast capacity under Gaussian channel is at most $O(\frac{\sqrt{n}}{n_s\sqrt{k}})$ when we have at least $n_s = \Omega(\log n)$ random multicast flows. Our result generalizes the unicast capacity for random networks using percolation theory.**

***Index Terms*—Capacity, Gaussian channel, multicast, percolation theory, scheduling, unicast, wireless ad hoc networks.**

I. INTRODUCTION

IN MANY applications, *e.g.*, wireless sensor networks, we often need an estimation on the (asymptotic) achievable throughput when we randomly deploy $\Theta(n)$ wireless nodes in a given region. The main purpose of this paper is to study the *asymptotic capacity* of large-scale *random wireless networks* where a large number of nodes are randomly placed in the deployment region, when we choose the *best* protocols for all layers. Due to spatial separation, several wireless nodes can transmit simultaneously, provided that these transmissions will not cause *destructive* wireless interferences to any of the simultaneous transmissions. To describe when a transmission is received successfully by its intended recipient, a number of interference models have been proposed and studied in the literature, which include the following models.

A. Protocol Interference Model (PrIM) [7]

In this model, a transmission by a node v_i is successfully received by an intended target v_j iff node v_j is sufficiently apart from the source of any other simultaneous transmission, *i.e.*, $\|v_k - v_j\| \geq (1+\eta)\|v_i - v_j\|$ for any simultaneously transmitting node $v_k \neq v_i$. Here, η is a constant depending on the environment.

B. Fixed-Power Protocol Interference Model (fPrIM)

Here, each node $v \in V$ has a fixed constant transmission range r and an interference range $R \geq r$. A node u can successfully receive a transmission from another node v iff: 1) $\|u - v\| \leq r$; and 2) there is *no* other node w with $\|w - u\| \leq R$ and node w is transmitting simultaneously with node v. Here, $\|w - u\|$ is the Euclidean distance between w and u.

C. Physical Interference Model(PhIM)

At any time, given a set of simultaneously transmitting nodes $A = \{u_1, u_2, \ldots, u_a\}$, a node v can successfully receive data from a sender $u \in A$ iff $\mathrm{SINR} = \frac{P_u \cdot \ell(u,v)}{N_0 + \sum_{i=1}^{a} P_{u_i} \ell(u_i, v)} \geq \sigma$. Here, σ is a threshold for SINR, P_{u_i} is the transmission power of node u_i, $0 < \ell(u_i, v) \leq 1$ is the path loss of signal propagation, and $N_0 > 0$ is the variance of background noise.

D. Gaussian Channel Model (GCM)

Given a set of simultaneously transmitting nodes $A = \{u_1, u_2, \ldots, u_a\}$, a node v can successfully receive data from a sender u at a data rate $\leq B \log(1 + \mathrm{SINR})$, where $\mathrm{SINR} = \frac{P_u \cdot \ell(u,v)}{N_0 + \sum_{i=1}^{a} P_{u_i} \ell(u_i, v)}$ and B is the bandwidth.

In the first three of the preceding models (PrIM, fPrIM, PhIM), when the transmission is successful, each wireless node can transmit at W bits/second over a common wireless channel. The unicast capacity for large-scale random wireless networks has been extensively studied. The groundbreaking work by Gupta and Kumar [7] has shown that: 1) for large-scale random networks of n nodes inside a unit square, the asymptotic per-flow unicast capacity with n random flows is $\Theta(W/\sqrt{n \log n})$ under fPrIM; 2) for networks where nodes are *arbitrarily* located (not necessarily randomly placed) in a unit square, when each node wishes to communicate to a random destination located at a nonvanishingly small distance

Manuscript received September 24, 2008; revised February 17, 2009 and September 07, 2009; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Sarkar. First published December 28, 2009; current version published August 18, 2010. This work was supported in part by NSF CNS-0832120, the National Natural Science Foundation of China under Grant 60828003, the National Basic Research Program of China (973 Program) under grants 2006CB30300 and 2010CB328100, the National High Technology Research and Development Program of China (863 Program) under Grant 2007AA01Z180, the Natural Science Foundation of Zhejiang Province under Grant Z1080979, the RGC under Grant HKBU 2104/06E, and the CERG under Grant PolyU-5232/07E.

X.-Y. Li is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA, and also with the Institute of Computer Application Technology, Hangzhou Dianzi University, Zhejiang 310018, China (e-mail: xli@cs.iit.edu).

Y. Liu is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (email: liu@cse.ust.hk)

S. Li is with Department of Computer Science, Princeton University, Princeton, NJ 08540 USA.

S. Tang is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: stang7@iit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2009.2037431

away, the amount of information that can be exchanged by each source–destination pair must go to zero, as $n \rightarrow \infty$, at least at rate $\Theta(W/\sqrt{n})$ under PrIM or PhIM. This result was originally proved as the consequences of the interference model used. It has later been extended to hold in a more general information theoretic setting [28]. Gupta and Kumar [7] also showed that when nodes are randomly located in a unit square area, each source–destination pair can achieve a bit rate only of order $\Theta(1/\sqrt{n \log n})$ when fPrIM or PhIM models are used. Under Gaussian channel model, using multihop transmission, pairwise coding and decoding at each hop, and a time-division multiple access (TDMA) scheme, Franceschetti *et al.* [3] show that a rate $\Omega(1/\sqrt{n})$ is achievable in networks of randomly located nodes. Then, they consequently claimed that there is no gap between the capacity of randomly located and arbitrarily located nodes, at least up to a constant scaling. Observe that these two results [3], [7] used two different channel models.

In this paper, we will concentrate on the asymptotic *multicast capacity* of random wireless networks. Our result will show how the multicast capacity scales with the number of nodes in the network or scales with the size of multicast group. Multicast capacity of random networks has been investigated recently. Using fixed-power protocol interference model fPrIM, Li *et al.* [15] showed that when there are n_s multicast flows and each multicast flow will have k randomly chosen receivers, the per-flow multicast capacity of n_s flows for random networks is of order $\Theta(\frac{W\sqrt{n}}{n_s\sqrt{k \log n}})$ when $k = O(n/\log n)$, and is of order $\Theta(W/n_s)$ when $k = \Omega(n/\log n)$.

For presentation simplicity, we assume that there is only one channel in the wireless networks. As always, we assume that the packets are sent from node to node in a multihop manner until they reach their final destinations. Unlike the PrIM, fPrIM, and PhIM models, there is no upper bound on the distance between the sending node and the receiving node in Gaussian channel model. The packets could be buffered at intermediate nodes while waiting for transmission. Intermediate nodes can only store and forward packets (no other operations such as network coding are allowed here). We assume that the buffer is large enough so packets will not get dropped by any intermediate node. We leave it as a future work to study the scenario when network coding is permitted and the buffers of intermediate nodes are bounded by some values. In some results, we assume that every intermediate node has an infinite buffer size. For most of the results presented here, the worst delay of the routing is not considered, *i.e.*, the delay in the worst case could be arbitrarily large for some results.

E. Our Main Contributions

This paper shows that a per-flow multicast rate $\Theta(1/\sqrt{nk})$ is achievable in networks of n randomly located nodes in a square region $\mathcal{B}_n = \sqrt{n} \times \sqrt{n}$. Specifically, we will prove the following main theorems.

Theorem 1: When $k \leq \theta_1 \frac{n}{(\log n)^{2\alpha+6}}$ and $n_s \geq \theta_2 n^{1/2+\beta}$ for some constants θ_1, θ_2 and any positive real number β, with high probability (*w.h.p.*),[1] each multicast source node can send data to all its intended receivers with rate at least

$$\lambda_{k,\mathcal{S}}(n) \geq c_8 \frac{\sqrt{n}}{n_s\sqrt{k}}. \quad (1)$$

Here, c_8 is a constant depending on $\alpha > 2$, θ_1, and θ_2.

Observe that the results from [27] indicate that the throughput order in Theorem 1 is achievable under the physical model, which can always serve as a lower bound to the Gaussian channel model. For completeness of presentation, we outline our proof of Theorem 1 here.

In terms of capacity upper bound, we proved the following.

Theorem 2: Under Gaussian channel model, the per-session multicast throughput for $n_s = \Theta(n)$ random flows in random networks in $\mathcal{B}_n$ is at most of order

$$\begin{cases} O\left(\frac{1}{\sqrt{kn}}\right), & \text{when } k : \left[1, \frac{n}{(\log n)^{\alpha}}\right] \\ O\left(\frac{1}{k(\log n)^{\frac{\alpha}{2}}}\right), & \text{when } k : \left[\frac{n}{(\log n)^{\alpha}}, n\right]. \end{cases} \quad (2)$$

Here, we use notation $k : [f(n), g(n)]$ to denote that $k = \Omega(f(n))$ and $k = O(g(n))$. Our results imply that for multicast under Gaussian channel model, if only relay and forwarding is allowed, the achievable per-session rate is asymptotically proportional to $\Theta(\frac{\sqrt{n}}{n_s\sqrt{k}})$ when $k = O(\frac{n}{(\log n)^{6+2\alpha}})$. The increase in the number of receivers will only decrease the throughput in the order of $1/\sqrt{k}$ for two-dimensional wireless networks. Observe that we do not know whether the boundary on k is tight such that the achievable per-session multicast rate is of order $\Theta(\frac{\sqrt{n}}{n_s\sqrt{k}})$. We think that the boundary most likely is not tight, and we want to know what is the tight asymptotic largest k such that this rate is still achievable. Recall that for the protocol model, Li *et al.* [15] derived a tight bound on k when two regimes of multicast capacity are separated: $k = O(\frac{n}{\log n})$ and $k = \Omega(\frac{n}{\log n})$. When $k = \Omega(\frac{n}{\log n})$, in protocol model, they [15] showed that, *w.h.p.*, a constant fraction of cells (with constant side length) will have receivers, thus, multicast is asymptotically the same as broadcast. We conjecture that $\Theta(\frac{n}{\log n})$ will also be a separation point on the value k in deriving different capacity regimes for multicast under Gaussian channel model. Also, notice that the hidden constants in all our formulas are not tight. A more careful analysis will further narrow the difference between the asymptotic upper bound and asymptotic lower bound on the capacity.

Compared to [15] and [23], studying the multicast capacity with Gaussian channel model requires new technical insights. Our result is derived based on the highway system that can be formed by use of percolation theory. The upper bound on asymptotic per-flow unicast capacity implied by Theorem 2 (when $k = 2$) shows that the unicast capacity achieved by [3] is indeed asymptotically optimal and thus finally closes the gap between the upper and lower bounds of unicast capacity when Gaussian link model is used.

The rest of the paper is organized as follows. In Section II, we briefly describe the network and system model used. Our routing strategy that can achieve asymptotic optimal multicast capacity is presented in Section III. We present the theoretic analysis in Section IV and present a matching upper bound for asymptotic per-flow multicast capacity in Section V when the number of receivers k is small. We review the related work in Section VI and conclude the paper in Section VII.

[1]Here, an event is said to happen with high probability if, for any $\epsilon \in (0, 1)$, there is an integer N (typically $N = 1/\epsilon$) such that for any random network of size $\geq N$, the event happens with probability $\geq 1 - \epsilon$.

II. Network and System Model

Consider a square region $\mathcal{B}_n$ of side length $\sqrt{n}$. We randomly place a number of nodes inside this square region by use of Poisson distribution with rate $\varrho = 1$, *i.e.*, the probability that a region $Z \subseteq \mathcal{B}_n$ has $i \geq 0$ nodes is $\frac{e^{-\varrho|Z|}(\varrho|Z|)^i}{i!}$. Here, $|Z|$ is the area of the region Z. Assume that each node will transmit at a constant power P, and node v_j receives the transmitted signal from v_i with power $P \cdot \ell(d(v_i, v_j))$, where $d(v_i, v_j)$ is the Euclidean distance between v_i and v_j, and $\ell(x)$ is the transmission loss during a path of length x. In this paper, we consider the attenuation function

$$\ell(x) = \min\{1, x^{-\alpha}\}$$

where the constant $\alpha > 2$. In a Gaussian channel model, the rate of a transmission from node v_i to node v_j is

$$\begin{aligned} R(v_i, v_j) &= B \log\left(1 + \frac{S(v_i, v_j)}{N_0 + I(v_i, v_j)}\right) \\ &= B \log\left(1 + \frac{P \cdot \ell(d(v_i, v_j))}{N_0 + \sum\limits_{k \neq i, v_k \in \mathcal{A}} P \cdot \ell(d(v_k, v_j))}\right) \end{aligned}$$

where $\mathcal{A}$ is the set of nodes transmitting simultaneously with node v_i, B is the channel bandwidth, $N_0 > 0$ is the variance of background noise, $I(v_i, v_j)$ is the total interference at the receiving node v_j when v_i is communicating with v_j, and $S(w, v)$ is the strength of signal (sent by w and received at v). When a node v_i simultaneously sends data to a set of receivers $\mathcal{D}$, the data rate that it can communicate is $R(v_i, \mathcal{D}) = \min_{v_j \in \mathcal{D}} R(v_i, v_j)$.

Assume that there are n_s multicast sessions. We randomly choose n_s nodes to be the sources of the multicast sessions. For each source node, we will choose $k-1$ nodes to be its intended receivers. The source nodes and their receivers are chosen using the the process described in Algorithm 1.

Algorithm 1 Process for selecting n_s multicast sessions

for $i \leftarrow 1, 2, \ldots, n_s$ **do**
  for $j \leftarrow 1, 2, \ldots, k$ **do**
    Randomly choose a point $p_{i,j}$ in $\mathcal{B}_n$.
    Choose a node $v_{i,j}$ from V that is closest to $p_{i,j}$
  end for
  Let $v_{i,1}$ be a source node and $v_{i,2}, v_{i,3}, \ldots, v_{i,k}$ be its intended receivers.
end for

In Algorithm 1, different multicast sessions may have the same source, and two receivers of a multicast session may be the same. A source node may also be an intended receiver of itself. These may confuse us when considering the multicast rate. Therefore, it is necessary to clarify them. If two receivers of a multicast session are the same, i.e., $v_{i,j_1} = v_{i,j_2}$, we can simply remove one of them. To notice that, a node can transmit data to itself with an arbitrary large rate. However, things are different when considering the set of n_s sources. If the sources of two multicast sessions are the same, we must treat them separately. Notice that both the transmitted data and the intended receivers of the two multicast sessions are different. We cannot combine the receivers of these two multicast sessions together either. One reason we choose the sources and receivers for each multicast session using Algorithm 1 is that we need the multicast sessions to be *independently* chosen when we analyze the achieved multicast capacity by our protocol using Vapnik–Chervonenkis (VC) dimension and VC theorem.

Given a random wireless network of n nodes and the set $\mathcal{S}$ of $n_s = |\mathcal{S}|$ source nodes, let $\lambda_{\mathcal{S}} = (\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_{n_s-1}}, \lambda_{i_{n_s}})$ be the *rate vector* of the multicast data rate of all n_s multicast sessions. Here, λ_{i_j} is the data rate of node $v_{i_j} \in \mathcal{S}$, for $1 \leq j \leq n_s$. In other words, we do *not* assume that all nodes will serve as the source of a multicast session. When given a *fixed* network $G = (V, E)$, where the node positions of all nodes V, the set $\mathcal{S}$ of n_s source nodes, the set of receivers U_i for each source node v_i, and the multicast data rate λ_i for each source node v_i are all fixed, we first define what is a feasible rate vector λ for the network G. A multicast rate vector $\lambda_{\mathcal{S}}$ bits/s is *feasible* if there is a spatial and temporal scheme for scheduling transmissions such that by operating the network in a multihop fashion and buffering at intermediate nodes when awaiting transmission, every node v_i can send λ_i bits/s on average to its chosen $k-1$ destination nodes. That is, there is a $T < \infty$ such that in every time interval (with unit seconds) $[(i-1) \cdot T, i \cdot T]$, every node $v_i \in \mathcal{S}$ can send $T \cdot \lambda_i$ bits to its corresponding $k-1$ receivers U_i *w.h.p.*

The total throughput of such feasible rate vector for multicast is defined as $\Lambda_{k,\mathcal{S}}(n) = \sum_{v_i \in \mathcal{S}} \lambda_i$. The average per-flow multicast throughput is $\lambda^a_{k,\mathcal{S}}(n) = \frac{\sum_{v_i \in \mathcal{S}} \lambda_i}{n_s}$. The minimum per-flow multicast throughput is $\lambda_{k,\mathcal{S}}(n) = \min_{v_i \in \mathcal{S}} \lambda_i$, where k is the total number of nodes in each multicast session, including the source node. When $\mathcal{S}$ is clear from the context, we drop $\mathcal{S}$ from our notations. When we mention *per-flow multicast capacity*, hereafter we mean the minimum per-flow multicast capacity, if not explained otherwise. An aggregated multicast throughput $\Lambda_k(n)$ bits/s is *feasible* for n_s multicast sessions (each session with k terminals) if there is a rate vector $\lambda_{\mathcal{S}} = (\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_{n_s-1}}, \lambda_{n_s})$ that is feasible and $\Lambda_k(n) = \sum_{v_i \in \mathcal{S}} \lambda_i$. Similarly, we say $\lambda_k(n) = \min_{v_i \in \mathcal{S}} \lambda_i$ is a feasible per-flow multicast throughput.

We say that the *multicast capacity per flow* of a class of random networks is of order $\Theta(f(n))$ bits/s if there are deterministic constants $c > 0$ and $c < c' < +\infty$ such that

$$\lim_{n \to \infty} \mathbf{Pr}\left(\lambda_k(n) = cf(n) \text{ is feasible}\right) = 1$$
$$\liminf_{n \to \infty} \mathbf{Pr}\left(\lambda_k(n) = c' f(n) \text{ is feasible}\right) < 1.$$

Here, the probability is computed using *all* possible random networks formed by n nodes distributed in a square $\mathcal{B}_n$. We will study the per-flow multicast capacity under Gaussian channel model instead of the fPrIM used in [15] and [23].

III. Our Multicast Routing Solution

In this section, we will first present several technical lemmas that will be used in our latter analysis. Then, we briefly review the highway system proposed in [3] and present our multicast method based on the highway system. We finally analyze the performance of our multicast method.

A. Technical Lemmas

We first present some technical lemmas that are essential for the analysis of asymptotic multicast capacity. Our first lemma shows that if the fixed range protocol model exclusion rules are respected, then some predetermined rate is achievable on each active link under the Gaussian channel model. Later, we will present our routing and scheduling, where these exclusion rules are respected for nodes in the highway system.

Lemma 3: At any time, assume that for *any* receiver v_i (and its sender s_i), the following conditions are satisfied:

- C_1: $\forall v_i$, the Euclidean distance $\|v_i s_i\| \leq r$;
- C_2: for any other sender s_k, $k \neq i$, the Euclidean distance between s_k and v_i is at least R with $R > r$.

Then, each receiver can receive at rate at least

$$B \log\left(1 + \frac{P \cdot \ell(r)}{N_0 + c_1 P (R - r)^{-\alpha}}\right)$$

where c_1 is a constant only depending on α.

See [17, Lemma 3] for the proof of this lemma. Observe that Lemma 3 still holds when a sender has multiple receivers. The lemma still holds, with a different constant data rate, if at any time slot every active link has a length at most r, and every pair of senders is separated by at least a distance $R_0 > 0$.

One may argue that, with Lemma 3, we can directly use the routing methods in [15] and [16] to get the achievable multicast rate under Gaussian channel model. In [15] and [16], it is assumed that all nodes have a transmission range r and interference range R, which are fixed constants. For the network model studied here, using a constant transmission range cannot get a connected network *w.h.p.*, due to results in [22]. Actually, to get a connected network *w.h.p.*, the transmission range of all nodes should be set as at least $\Theta(\sqrt{\log n})$. Thus, the assumption that each link (when no other active links exist) has a constant data rate W used in [15] and [16] does not hold anymore: The data rate W achievable by the worst links in a connected network under Gaussian channel model is of order $W = O(\frac{1}{(\log n)^{\alpha/2}})$, and even other links are not active. Thus, the data rate achievable by directly applying the routing and scheduling methods in [15] and [16] to the network model here (under Gaussian channel model) is only of order $\Theta(W \cdot \frac{\sqrt{n}}{n_s \sqrt{k \log n}}) = \Theta(\frac{\sqrt{n}}{n_s \sqrt{k}} \cdot \frac{1}{(\log n)^{\frac{(\alpha+1)}{2}}})$, when $k = O(\frac{n}{\log n})$. This rate is only $(\log n)^{-\frac{(\alpha+1)}{2}}$ fraction of the rate achieved by our methods presented later, when $k = O(\frac{n}{\log^{6+2\alpha} n})$.

Lemma 4: For $\gamma \geq 1$, if we partition $\mathcal{B}_n = [0, \sqrt{n}] \times [0, \sqrt{n}]$ into at least $\tau_1 \frac{n}{\log^\gamma n}$ subsquare regions (called **cell**) of area at most $\tau_2 \log^\gamma n$, then *w.h.p., every* region contains at most $2\tau_2 \log^\gamma n$ nodes. Here, τ_1 and $\tau_2 > \frac{1}{(2 - \log e)}$ are constants.

See [17, Lemma 4] for the proof of this lemma. Observe that when $\gamma \geq 1$, $\Pr(A_n) \leq \frac{\tau_1}{n^{\tau_2 \log(4/e) - 1} \log n}$.

Lemma 5: If we partition $\mathcal{B}_n$ into regions of area $\geq a \ln n$ (for $a \geq 1$), then *w.h.p.*, *every* region contains at least one node.

Proof: Let A_n be the event that some region is empty of nodes. Then, $\Pr(A_n) \leq \lceil \frac{n}{a \ln n} \rceil e^{-a \ln n} = \lceil \frac{n}{a \ln n} \rceil \frac{1}{n^a} \to 0$ as n tends to infinity. Lemma then follows. ■

Observe that Lemmas 4 and 5 still hold when nodes are produced by uniform random distribution.

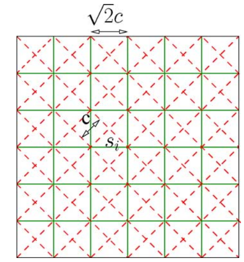


Fig. 1. Construction of the bond percolation model.

B. Constructing Highway System Using Percolation Theory

Our routing strategy is built upon the highway system developed in [3]. We first review the highway system defined in [3]. To begin the construction of the highway system, we partition the deployment box $\mathcal{B}_n$ into cells of a constant side length c, as depicted in Fig. 1.

In Fig. 1, let $N(s_i)$ be the number of random nodes inside a cell s_i. By appropriately choosing c, we can arrange that the probability that a square contains at least a Poisson node is as high as we want. Indeed, for all i, we have $p \equiv \Pr(N(s_i) \geq 1) = 1 - e^{-c^2}$. We say that a square is *open* if it contains at least one node, and *closed* otherwise. Notice that squares are open (and closed) with a probability p (and $1 - p$), independently of each other, because the nodes are produced by Poisson distribution. Thus, percolation theory can be applied here. This model is then mapped into a discrete edge-percolation model on the square grid as follows.

We associate an edge to each cell, traversing it diagonally, as depicted by horizontal and vertical segments in Fig. 1. The edge is said to be either open or closed according to the state of the corresponding cell. We then obtain a grid G_n of horizontal and vertical edges, each edge being open, independently of all other edges, with probability p. A path of G_n is said to be *open* if it contains only open edges. Observe that an open path implies that we have a routing path (by selecting one node from each open cell and connecting nodes from adjacent open cells) such that every link on the path has length at most a constant $\sqrt{5}c$. Thus, the data rate achievable by this path is of a constant value (depending on c) from Lemma 3, using a TDMA scheduling of nodes [3]. Note that when constant c is large enough, the preceding construction produces open paths that cross the entire network area.

Denote the number of edges composing the side length of $\mathcal{B}_n$ by $m = \frac{\sqrt{n}}{c\sqrt{2}}$, where c is rounded up such that m is an integer. By Theorem 22, we can choose c large enough such that, *w.h.p.*, there are $\Omega(m)$ paths crossing $\mathcal{B}_n$ from left to right. These paths can be grouped into disjoint sets of paths: each group has $\lceil \delta \log m \rceil$ paths, crossing a rectangle of width m and height $\kappa \log m - \epsilon_m$, for all $\kappa > 0$, δ small enough, and a vanishingly small ϵ_m so that the side length of each rectangle is an integer. See Fig. 2 for illustration. The same is true if we divide the area into vertical rectangles and look for paths crossing the area from bottom to top. Using the union bound, they [3] conclude that there exist both horizontal and vertical disjoint paths *w.h.p*. These paths form a backbone called the *highway system* [3].

We then slice each horizontal rectangle (of width m and height $\kappa \log m - \epsilon_m$) into horizontal strips of constant height

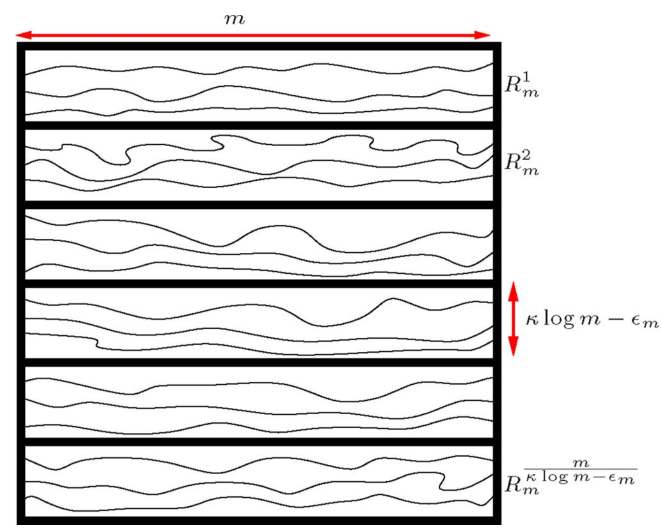


Fig. 2. There exist a number of horizontal crossing paths in B_m.

h. By choosing h appropriately, we can guarantee that there are at least the same paths as strips in every strip. Similarly, we can divide the vertical rectangle into vertical strips. We let $H = \kappa \log m - \epsilon_m$ be the height of the horizontal rectangles (or the width of the vertical rectangles), h be the height of the strips (or the width of the vertical strip), $J = \sqrt{n}/H$ be the number of horizontal (vertical) rectangles, and $L = H/h$ be the number of horizontal (vertical) strips in a horizontal (vertical) rectangle. As there are at least the same horizontal (vertical) highways as the strips in a horizontal (vertical) rectangle, L node-disjoint horizontal crossing highways can be chosen in each rectangle. In all, we choose $M = J \times L$ horizontal (vertical) highways.

Let $\Pi_1, \Pi_2, \ldots, \Pi_M$ be the M horizontal highways such that $\Pi_{(i-1)L+j} (1 \leq i \leq J, 1 \leq j \leq L)$ is a highway in the ith rectangle. We also let $\pi_{i,j}$ be the jth node in the ith horizontal highway. Therefore, a highway Π_i can be denoted by a list of nodes, i.e, $\Pi_i = (\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,s_i})$. Similarly, we use $\Phi_1, \Phi_2, \ldots, \Phi_M$ to denote the M vertical highways, where $\Phi_i = (\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,t_i})$. In this paper, we propose the following definition that will be used in our proofs later.

Definition 1: We call a horizontal (vertical) highway $\Pi_i = (\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,s_i})$ (or $\Phi_i = (\phi_{i,1}, \phi_{i,2}, \ldots, \phi_{i,t_i})$) *almost-straight* if there does not exist j_1, j_2 such that $1 \leq j_1 < j_2 \leq s_i$(or t_i) and $X(\pi_{i,j_1}) > X(\pi_{i,j_2}) + 2H$ (or $Y(\phi_{i,j_1}) > Y(\phi_{i,j_2}) + 2H$). Here, $X(p)$ and $Y(p)$ are the x-coordinate (from left to right) and y-coordinate (from up to down) of point p, respectively.

Essentially, almost-straight highways (called *legal* in [17]) are highways that will go backward at most of distance $2H$. The existence of almost-straight highways will ensure that: 1) the Euclidean minimum spanning tree can be approximated by using highways: 2) the capacity achievable by the highway system is large. In [17], we proved the following theorem.

Theorem 6: If we find a set of M horizontal highways and M vertical highways using the percolation method, we can find a set of M almost-straight horizontal highways and M almost-straight vertical highways.

In the rest of the paper, we will always use the almost-straight highways.

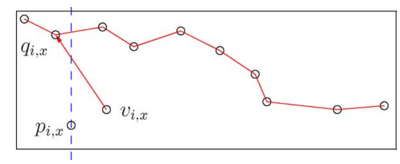


Fig. 3. Choose $q_{i,x}$ for $v_{i,x}$ where the path is a highway.

C. Schedule the Multicast Tasks

We now are ready to describe our multicast method (summarized in Algorithm 2). The proposed solution is based on multihop routing and exploits the formation of paths percolating across the network. As in [3], we divide the nodes into disjoint sets that cross the network area. These sets form a "highway system" of nodes (called *stations* sometimes) that can carry information across the network at constant rate using short hops. The rest of the nodes access the highway system using single hops of longer lengths.

Algorithm 2 Find a Euclidean Spanning Tree for k points

Input: $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k}\}$
Output: A Euclidean tree spanning P_i, denoted as $EST(P_i)$
Algorithm:

```
t ← the minimum integer such that 4^t ≥ k;
𝒫 ← P_i; and ℰ ← ∅;
for g ← t − 1, ..., 1, 0 do
    Divide ℬ_n into 2^g × 2^g cells, each with size a/2^g × a/2^g;
    for each cell of size a/2^g × a/2^g do
        if the cell contains s ≥ 2 points in 𝒫 then
            Randomly choose a point p_{i,x} ∈ 𝒫 in cell;
            for any other point p_{i,y}(y ≠ x) in this cell do
                ℰ ← ℰ ⋃ {p_{i,x}p_{i,y}}; 𝒫 ← 𝒫 − {p_{i,y}};
            end for
        end if
    end for
end for
Output ℰ as the edges of EST(P_i).
```

Our multicast protocol (Algorithm 3) contains two kinds of hops: the constant-length hop in the highway system and the longer hop connecting a receiver $v_{i,x}$ to some entry node $q_{i,x}$ in the highway. We will then perform multicast (using multicast tree) to these entry nodes in the highway. To transmit data through the multicast tree, we divide our communication strategy into three separate phases:

1) In the first phase, every nonstation node $v_{i,x}$ exchanges its data with some station $q_{i,x}$ in the highway system (we call the nodes in the highway system *stations*) using a single-hop communication; see Fig. 3.
2) In the second phase, data is transmitted through highways using station nodes that are part of some special Euclidean spanning tree constructed.
3) In the third phase, data is forwarded directly to the destination nodes from the nodes of the highway system.

In the rest of our analysis, we typically will not distinguish the first phase and the third phase. In the following, we take all the n_s multicast sessions into consideration and analyze the date rate per multicast session of the two phases separately.

We first describe our method (Algorithm 2) to construct a Euclidean spanning tree of a set P_i of k points. We have to point out that our method will not necessarily construct a Euclidean minimum spanning tree of these k points. Assume that the set P_i of k points is located in a square region $[0, a] \times [0, a]$. Our method for constructing a Euclidean spanning tree will first divide the region into cells (with side length $a/2^{t-1}$ for $t = \lceil \log_4 k \rceil$). This cells are called level $t-1$ cell. Similarly, we can define level g cells with side length $a/2^g$. Originally, all nodes are representant nodes in level $t-1$. If a level i cell contains some representant nodes, we randomly pick one (as the representant node to upper level $i-1$) and build edges from all other representant nodes in this cell to the randomly picked node. We will show that the Euclidean length of the constructed tree is of same order as the Euclidean length of Euclidean minimum spanning tree.

Algorithm 3 Build a multicast tree using highway

Input:

1) $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k}\}$ and $EST(P_i)$ generated from Algorithm 2,
2) $V_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,k}\}$ generated by Algorithm 1,
3) M horizontal highways $\Pi_1, \Pi_2, \ldots, \Pi_M$ and M vertical highways $\Phi_1, \Phi_2, \ldots, \Phi_M$ as described previously.

Output: A multicast tree spanning V_i, denoted as $MT(V_i)$.

for $x \leftarrow 1, 2, \ldots, k$ **do**
  Suppose $p_{i,x}$ is in the z_x-th horizontal strip;
  Let $q_{i,x}$ be the node from Π_{z_x} which is closest to the vertical line drawn from $p_{i,x}$ (see **Fig. 3**);
  ▷ $q_{i,x}$ will relay data for $v_{i,x}$.
end for
for each edge $\overline{p_{i,x}p_{i,y}}$ in $EST(P_i)$ **do**
  Suppose $q_{i,x} = \pi_{z_x,u_0}$, and $q_{i,y} = \pi_{z_y,u_5}$;
  if $z_x = z_y$ **then**
    $E(q_{i,x}, q_{i,y}) \leftarrow (\pi_{z_x,u_0}, \pi_{z_x,u_0 \pm 1}, \ldots, \pi_{z_x,u_5})$.
  else
    Suppose $p_{i,x}$ is on the w_x-th vertical strip.
    Find a station π_{z_x,u_1} in Π_{z_x} and a station ϕ_{w_x,u_2} in Φ_{w_x} such that $d(\pi_{z_x,u_1}, \phi_{w_x,u_2}) \leq \sqrt{5}c$;
    Find a station ϕ_{w_x,u_3} in Φ_{w_x} and a station π_{z_y,u_4} in Π_{z_y} such that $d(\phi_{w_x,u_3}, \pi_{z_y,u_4}) \leq \sqrt{5}c$;
    $E_1(q_{i,x}, q_{i,y}) \leftarrow (\pi_{z_x,u_0}, \pi_{z_x,u_0 \pm 1}, \ldots, \pi_{z_x,u_1})$;
    $E_2(q_{i,x}, q_{i,y}) \leftarrow (\phi_{w_x,u_2}, \phi_{w_x,u_2 \pm 1}, \ldots, \phi_{w_x,u_3})$;
    $E_3(q_{i,x}, q_{i,y}) \leftarrow (\pi_{z_y,u_4}, \pi_{z_y,u_4 \pm 1}, \ldots, \pi_{z_y,u_5})$;
    $E(q_{i,x}, q_{i,y}) \leftarrow E_1(q_{i,x}, q_{i,y}) \propto E_2(q_{i,x}, q_{i,y}) \propto E_3(q_{i,x}, q_{i,y})$; ▷ See **Fig. 4** for illustration, $\propto$ means concatenation of paths. ▷ Here $E(q_{i,x}, q_{i,y})$ is a path in the highway connecting $q_{i,x}$ and $q_{i,y}$ (See **Fig. 4**).
  end if
end for
Let $MT'(V_i)$ be the set of edges that covered by any path $E(q_{i,x}, q_{i,y})$, union the set $\{\overline{q_{i,x}v_{i,x}} \mid 1 \leq x \leq k\}$.
$MT'(V_i)$ is a connected graph that covers V_i. We can remove redundant edges to get a multicast tree, denoted as $MT(V_i)$.

After we construct the Euclidean spanning tree as guideline for routing, we then describe our method (Algorithm 3) to construct the actual multicast tree for a multicast composed of nodes $V_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,k}\}$, which are generated by Algorithm 1. To ensure that the multicast trees are *independent* of each other for different multicast sessions, we actually will first build a multicast tree for points $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k}\}$, $1 \leq i \leq n_s$. For each edge $\overline{p_{i,x}p_{i,y}}$ in $EST(P_i)$, we will first find the closest entrance nodes $q_{i,x}$, $q_{i,y}$ for points $p_{i,x}$, $p_{i,y}$ and connect nodes $q_{i,x}$, $q_{i,y}$ using a manhattan-like path (formed of three components $E_1(q_{i,x}, q_{i,y})$, $E_2(q_{i,x}, q_{i,y})$, $E_3(q_{i,x}, q_{i,y})$) in the highway. We then send data from $p_{i,x}$ to $q_{i,x}$ using multihops, then use the links in the highway to forward the data, at last forwarding data from $q_{i,y}$ to $p_{i,y}$ using multihops.

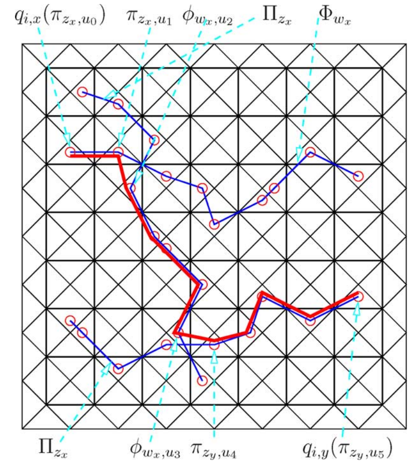


Fig. 4. A path connecting $q_{i,x}$ and $q_{i,y}$ contains three highway segments: the horizontal one from $q_{i,x}$ to π_{z_x,u_1}, the vertical one from ϕ_{w_x,u_2} to ϕ_{w_x,u_3}, and the horizontal one from π_{z_y,t_4} to $q_{i,y}$. These three segments are connected by shortcuts, $\pi_{z_x,u_1}\phi_{w_x,u_2}$ and $\phi_{w_x,u_3}\pi_{z_y,u_4}$ of length at most $\sqrt{5}c$.

We will first study the capacity that can be supported by the network, assuming that P_i forms nodes in a multicast session. In our study, we will use VC dimension and VC theorem, which require the multicast sessions to be independent, which is true if P_i are multicast terminals. For actual multicast of V_i, we will then directly connect each node $v_{i,j}$, $1 \leq j \leq k$, to the entrance node, say $q_{i,j}$, in the highway system. We will show that the capacity is not reduced asymptotically.

We schedule the link transmissions using TDMA as in [3], [15], and [16]. We first divide the time into mega-slots. One mega-slot is then divided into two equal-sized groups of mini time slots. The first group of mini time slots will be reserved for nodes in the highway system, and the second group of mini time slots will be reserved for nodes to relay data to (or from) the highway system. We divide $\mathcal{B}_n$ into cells of side length c. Each time at most one node from a cell can transmit and at any time the transmitting nodes are separated by at least $t \geq 1$ cells. Thus, every cell will have a node that can transmit every t^2 mini time slots.

IV. Analysis of Achievable Capacity

We now analyze the per-flow multicast capacity achievable by our routing and scheduling protocol.

A. Data Rate of the First, Third Phase (Accessing Highway)

To notice that a receiver will have the same relay node from highways in all multicast sessions, our computation of the data rate from a node to its highway entrance station comprises two steps. In the first step, we only need to analyze the rate between receivers and their relay nodes. In the second step, we calculate how many multicast sessions a nonstation node v^* is covered by, which will imply the data rate achievable in first and third phase.

Lemma 7: In the first (and third) phase of the transmission, *w.h.p.*, for any $1 \leq i \leq n_s$ and for any $x(1 \leq x \leq k)$, the data rate achievable by our method between a terminal $v_{i,x}$ and the highway entrance station $q_{i,x}$ is $c_2(\log n)^{-\alpha-2}$ in both directions. Here, c_2 is a constant.

Proof: Notice that the node $p_{i,x}$ and $q_{i,x}$ are within the same rectangle with height H, and the horizontal distance between them is at most $\sqrt{2}c$. Then, the distance between $p_{i,x}$ and $q_{i,x}$ is at most $H + \sqrt{2}c$.

From Lemma 5, we can see w.h.p. there is at least one node in every region with area $\log n$. Thus, we could divide square $\mathcal{B}_n$ into squares with side length $(1+\xi_n)\sqrt{\log n}$, where ξ_n is the smallest positive number that $\frac{\sqrt{n}}{(1+\xi_n)\sqrt{\log n}}$ is an integer. It is easily seen that ξ_n tends to 0 when n tends to ∞. Since w.h.p. each square contains a node and $v_{i,x}$ is the closest node from the point $p_{i,x}$, the distance $d(p_{i,x}, v_{i,x})$ is at most $\sqrt{2}(1+\xi_n)\sqrt{\log n}$.

By adding the above two upper bounds, we can see that the distance between $v_{i,x}$ and $q_{i,x}$ is at most $H + \sqrt{2}c + \sqrt{2}(1+\xi_n)\sqrt{\log n} = \kappa \log m - \epsilon_m + \sqrt{2}c + \sqrt{2}(1+\xi_n)\sqrt{\log n}$. This is smaller than $2\kappa\log m$ for a sufficient large n. Note $m = \sqrt{n}/(c\sqrt{2})$.

We let $r = 2\kappa \log m$ and $R = 2r$. Then, by Lemma 3, the data rate $R(v_{i,x}, q_{i,x})$ that can be achieved between $v_{i,x}$ and $q_{i,x}$ is at least $B \log\left(1 + \frac{P\cdot\ell(r)}{(N_0+c_1P(R-r)^{-\alpha})}\right)$ when the condition C_2 of Lemma 3 is satisfied. This condition can be guaranteed by dividing the phase 1 into time slots. We partition the square $\mathcal{B}_n$ into a number of cells with length r and divide the phase 1 into 16 time slots such that within a time slot, any two cells that contain transmitting nodes is at least four cells away (see Fig. 5(a) for illustration). Thus, any two transmitting nodes are at least $3r$ away from each other. To make sure that at the same time there is at most onetransmitting node at each cell, each of the 16 time slots should be divided into smaller mini time slots. By Lemma 4, we can see $2r^2$ mini time slots is enough since, *w.h.p.*, each cell contains at most $2r^2$ nodes. Considering the number of mini time slots, *w.h.p.*, the data rate between each pair of $v_{i,x}$ and $q_{i,x}$ that we can achieve is at least

$$
\begin{aligned}
&B\log\left(1+\frac{P\cdot\ell(r)}{N_0+c_1P(R-r)^{-\alpha}}\right)/(16\times 2r^2)\\
&\geq (1-\varepsilon_1)BP\cdot r^{-\alpha}/(32N_0r^2)\\
&=(1-\varepsilon_1)\frac{BP}{32N_0}r^{-\alpha-2} \geq (1-\varepsilon_1)\frac{BP}{32N_0}\left((1+\varepsilon_2)\frac{\log n}{2}\right)^{-\alpha-2}\\
&\geq \frac{2^\alpha BP}{17N_0}(\log n)^{-\alpha-2}.
\end{aligned}
$$

The above inequality requires that n is sufficiently large. In the above inequality, ε_1 and ε_2 are positive numbers whose value we can set. In the above reasoning, we assigned each node a time slot, and thus v_{i_x} and $q_{i,x}$ will have separate time slots. Thus, the rates in both directions can achieve the lower bound. Setting $c_2 = \frac{2^\alpha BP}{17N_0}$ will finish our proof. ■

Now, we move to the second step. We need to show how many multicast sessions a node v^* may be part of. First, we consider the process $\mathcal{Q}$ for choosing one node v^*: randomly selecting a point q^* in $\mathcal{B}_n$, and let v^* be its nearest wireless node. Then, what is the probability that a node v^* is chosen in this process $\mathcal{Q}$? The following lemma gives the answer.

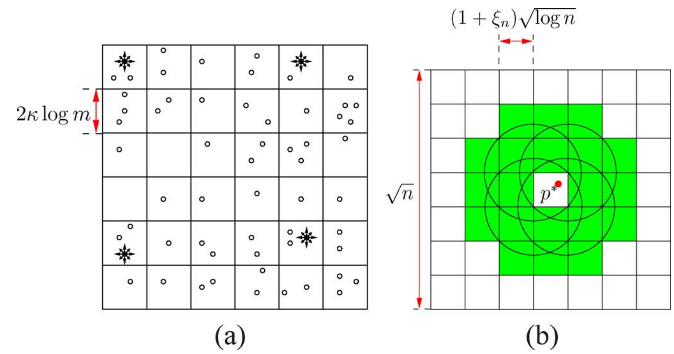


Fig. 5. (a) The cells that contain transmitting nodes are at least four cells away from each other, and each cell contains at most one transmitting node. The nodes with arrows represent transmitting nodes. (b) The cells where v^* may be located. p^* is located in the square in the center, and the shaded squares and the center square (total 21 squares) are the squares where v^* may be located *w.h.p.* The statement is also correct when we exchange the position of v^* and p^*.

Lemma 8: W.h.p., for any node v^*, the probability that a node v^* is chosen by process $\mathcal{Q}$ is at most $c_3\frac{\log n}{n}$ for a constant c_3.

Proof: This is exactly to compute the area of the regions in the Voronoi graph of the n nodes. In Lemma 5, we partition the square $\mathcal{B}_n$ into cells of side length $(1+\xi_n)\sqrt{\log n}$, and w.h.p. each cell contains at least one node. Considering a point p^* in a cell s, *w.h.p.*, its nearest node v^* must fall in s or the 20 cells around s [see Fig. 5(b)]. In other words, if v^* is in a cell s', p^* must fall in s' or the 20 cells around s'. Therefore, the probability that a node v^* is chosen by process $\mathcal{Q}$ is at most $21\frac{(1+\xi_n)^2\log n}{n}$. Since ξ_n tends to 0 as n tends to $+\infty$, it is smaller than $22\frac{\log n}{n}$ when n is sufficiently large. Therefore, if we let $c_3 = 22$, w.h.p., for any station v^*, the probability is at most $c_3\frac{\log n}{n}$. ■

Lemma 9: W.h.p., for any node v^*, the probability that a multicast session has v^* as a receiver is at most $c_3k\frac{\log n}{n}$.

Proof: Since the probability that a node v^* is chosen by process $\mathcal{Q}$ is at most $c_3\frac{\log n}{n}$, and v^* is chosen by a multicast session as receiver if v^* is chosen by at least one of k processes, the probability is at most $c_3k\frac{\log n}{n}$. ■

Lemma 10: In Algorithm 1, w.h.p., for any node v^*, the number of times that v^* is chosen by process $\mathcal{Q}$ as a multicast receiver is at most $3c_3n_sk\frac{\log n}{n}$ when $n_sk \geq n$.

Proof: Let A_n be the event that a node v^* is chosen by $\mathcal{Q}$ more than $3c_3n_sk\frac{\log n}{n}$ times. Let $p = c_3k\frac{\log n}{n}$, the probability that v^* is chosen as terminal of a multicast session. Then

$$
\begin{aligned}
\Pr(A_n) &\leq n_s\binom{n_s}{3n_sp}p^{3n_sp} \leq n_s\left(\frac{n_se}{3n_sp}\right)^{3n_sp}p^{3n_sp}\\
&= n_s\left(\frac{e}{3}\right)^{3n_sp} \leq n_s\left(n^{-3c_3(\log 3-1)}\right)^{n_sk/n} \to 0
\end{aligned}
$$

because $3c_3(\log 3 - 1) > 1$ and $n_sk \geq n$. ■

Lemma 11: W.h.p., there exists a constant $c_4 > 0$, the data rate that any multicast session can achieve in the first and third phase is at least $c_4\frac{\sqrt{n}}{n_s\sqrt{k}}$, if $k \leq \theta_1\frac{n}{\log^{2\alpha+6}n}$ and $n_s \geq \theta_2n^{1/2+\beta}$, where θ_1, θ_2 are special constants, and $\beta > 0$ is any positive real number.

Proof: When $n_s k \geq n$ and $k \leq \theta_1 \frac{n}{\log^{2\alpha+6} n}$, based on Lemmas 7 and 10, w.h.p., the data rate achievable per multicast session in the first and third phase is

$$R_1^1 \geq \frac{c_2(\log n)^{-\alpha-2}}{3c_3 n_s k \frac{\log n}{n}} = \frac{c_2}{3c_3}\frac{n(\log n)^{-\alpha-3}}{n_s k} \geq \frac{c_2}{3c_3}\left(\frac{n(\log n)^{-\alpha-3}}{n_s\sqrt{k}}\right) / \left(\sqrt{\theta_1 \frac{n}{\log^{2\alpha+6} n}}\right) = \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{\sqrt{n}}{n_s\sqrt{k}}.$$

When $n_s k < n$, the number of multicast sessions that will choose a given node as receiver is w.h.p. at most $3c_3 n \frac{\log n}{n} = 3c_3 \log n$. Then, when $n_s k < n$ and $n_s \geq \theta_2 n^{1/2+\beta}$, *w.h.p.*, the data rate that every multicast session can achieve in both first and third phases is

$$R_1^2 \geq \frac{c_2(\log n)^{-\alpha-2}}{3c_3 n \frac{\log n}{n}} \geq \frac{c_2}{3c_3}(\log n)^{-\alpha-3} \geq \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{n^{-\beta}}{\sqrt{k}} \geq \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{\sqrt{n}}{n_s\sqrt{k}}.$$

In all, *w.h.p.*, the data rate of any multicast session in the first phase is at least, when $k \leq \theta_1 \frac{n}{\log^{2\alpha+6} n}$ and $n_s \geq \theta_2 n^{1/2+\beta}$

$$R_1 \geq \min(R_1^1, R_1^2) \geq \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{\sqrt{n}}{n_s\sqrt{k}}.$$

The lemma then follows by setting $c_4 = \frac{c_2}{3c_3\sqrt{\theta_1}}$. ■

Note we assumed that $k \leq \theta_1 \frac{n}{\log^{2\alpha+6} n}$ and $n_s \geq \theta_2 n^{1/2+\beta}$. It is interesting to see if our results still hold for general k.

B. Capacity of the Highway System

We then study the capacity of the highway system for multicast. We begin our analysis on the spanning tree used for multicast constructed by Algorithm 2.

Lemma 12: In the second phase, the probability that a station node is covered by a multicast session is at most $c_5 \frac{\sqrt{k}}{\sqrt{n}}$ when $k \leq \theta_3 \frac{n}{\log^2 n}$, where c_5 and θ_3 are constants.

See the Appendix for the proof of the lemma. With Lemma 12, the following lemma is straightforward.

Lemma 13: When $k \leq \theta_3 \frac{n}{\log^2 n}$, for any station v^*, the expected number of multicast sessions that pass v^* is $\leq c_5 \frac{n_s\sqrt{k}}{\sqrt{n}}$.

Proof: Since the n_s multicast sessions are generated independently, multiplying the upper bound of the probability that v^* is covered by a multicast session by n_s will result in the upper bound of the expected number of covering multicast sessions. That is, $c_5 \frac{\sqrt{k}}{\sqrt{n}} \times n_s = c_5 \frac{n_s\sqrt{k}}{\sqrt{n}}$. ■

The preceding result only shows an upper bound on probability that a given node v^* is used by multicast sessions when v^* is given *a priori*. Next, we use VC theorem (Theorem 24) to give an upper bound on the number of multicast sessions that pass v^* for *every* possible node v^* in the highway system. Recall that we used n_s sets of *independently selected* k points to generate n_s multicast trees. Therefore, the input space should be the family of sets of k points, i.e., $[0, \sqrt{n}]^{2k}$. To notice that the output MT of Algorithm 3 is fixed for a fixed set of k points, we could set the universal input space $\mathcal{U}$ be the set of all possible output multicast trees of Algorithm 3. For each wireless station v^*, v^* is either covered or not covered by a tree T in $\mathcal{U}$. For a subset S of $\mathcal{U}$, we use $\mathcal{T}_S(v^*)$ to denote the set of trees from S that cover v^*. Let

$$\mathcal{C}_S = \{\mathcal{T}_{\mathcal{U}}(v^*) \mid v^* \text{ is a node in the highway system}\}.$$

Our objective is to compute the VC dimension VC-d$(\mathcal{C}_{\mathcal{U}})$ of $\mathcal{C}_{\mathcal{U}}$. Here, we simply use $\log_2 n$ as the upper bound of VC-d$(\mathcal{C}_{\mathcal{U}})$ due to the fact that there are at most n elements in $\mathcal{C}_{\mathcal{U}}$. Notice that a careful analysis can show that the VC dimension VC-d$(\mathcal{C}_{\mathcal{U}})$ is actually of order $\Theta(\log k)$[16].

Theorem 14: With high probability, for every station v^*, the number of multicast sessions that cover v^* is at most $c_6 \frac{n_s\sqrt{k}}{\sqrt{n}}$, when $k \leq \theta_3 \frac{n}{\log^2 n}$ and $n_s \geq \theta_2 n^{1/2+\beta}$, where c_6 is a constant to be specified and $\beta > 0$ is *any* positive real number.

Proof: Recall that in Lemma 12, the probability that a station v^* is covered by a random multicast session is at most $c_5 \frac{\sqrt{k}}{\sqrt{n}}$. Using VC theorem, with n_s multicast sessions

$$\Pr\left(\sup_{v^*}\left|\frac{\#\text{of sessions covering } v^*}{n_s} - c_5\frac{\sqrt{k}}{\sqrt{n}}\right| < \epsilon(n)\right) > 1 - \sigma(n)$$
$$\text{if } n_s \geq \max\left\{\frac{8d}{\epsilon(n)} \cdot \log\frac{13}{\epsilon(n)}, \frac{4}{\epsilon(n)}\log\frac{2}{\sigma(n)}\right\}.$$

If we set $\epsilon(n) = \frac{\sqrt{k}}{\sqrt{n}}$, $\sigma(n) = \frac{2}{n}$ and let $F(v)$ be the number of multicast sessions that use node v, we have

$$\Pr\left(\sup_{v^*}(F(v^*)) < (c_5+1)\frac{n_s\sqrt{k}}{\sqrt{n}}\right) > 1 - \frac{2}{n}$$
$$\text{if } n_s \geq \max\left\{\frac{8\sqrt{n}\log n}{\sqrt{k}} \cdot \log\frac{13\sqrt{n}}{\sqrt{k}}, \frac{4\sqrt{n}}{\sqrt{k}}\log n\right\} = \frac{8\sqrt{n}\log n}{\sqrt{k}} \cdot \log\frac{13\sqrt{n}}{\sqrt{k}}.$$

To guarantee the above lower bound for n_s for a large enough n, it is sufficient that $n_s \geq \theta_2 n^{1/2+\beta}$ for a constant $\beta > 0$. Let $c_6 = c_5 + 1$, and we finish the proof. ■

Lemma 15: W.h.p., the data rate of the second phase in any multicast session is at least $c_7 \frac{\sqrt{n}}{n_s\sqrt{k}}$, when $n_s \geq \theta_2 n^{1/2+\beta}$ and $k \leq \theta_3 \frac{n}{\log^2 n}$.

Proof: As the distance between two adjacent highway stations is at most $2\sqrt{2}c$, we can set $r = 2\sqrt{2}c$ and $R = 4\sqrt{2}c$ and apply Lemma 3. We do it in the similar way with the proof of Lemma 7. As there is at most one station in a square of size $c \times c$, we only need to divide the second phase into $\left(\lceil\frac{(R+r)}{c}\rceil + 1\right)^2 = 100$ time slots. Then, w.h.p., each station can send data to its adjacent stations (on the same highway) at rate at least $B\log\left(1+\frac{P\ell(2\sqrt{2}c)}{(N_0+c_3P(2\sqrt{2}c)^{-\alpha})}\right)/100 = \Theta(1)$.

In addition, w.h.p., every station in highway is covered by at most $c_6 \frac{n_s\sqrt{k}}{\sqrt{n}}$ multicast sessions when $k \le \theta_3 \frac{n}{\log^2 n}$. Therefore, each multicast session has a rate at least

$$R_2 \ge B \log\left(1+\frac{P\cdot \ell(2\sqrt{2}c)}{N_0+c_3P(2\sqrt{2}c)^{-\alpha}}\right)\Big/\left(100c_6\frac{n_s\sqrt{k}}{\sqrt{n}}\right)$$

$$=\frac{B}{100c_6}\log\left(1+\frac{P\cdot \ell(2\sqrt{2}c)}{N_0+c_3P(2\sqrt{2}c)^{-\alpha}}\right)\frac{\sqrt{n}}{n_s\sqrt{k}}.$$

Thus, if letting $c_7=(\frac{B}{100c_6})\log\left(1+\frac{P\cdot\ell(2\sqrt{2}c)}{(N_0+c_3P(2\sqrt{2}c)^{-\alpha})}\right)$, we get the result we need. ■

C. *Per-Flow Multicast Capacity of the System*

By combining the data rate in the two phases, we have the following.

Theorem 16: If $k \le \theta_1 \frac{n}{\log^{2\alpha+6} n}$ and $n_s \ge \theta_2 n^{1/2+\beta}$, *w.h.p.,* the per-flow multicast rate is at least $c_8 \frac{\sqrt{n}}{n_s\sqrt{k}}$, where $c_8 = (\frac{1}{2})\min\{c_4, c_7\}$.

Proof: When $k \le \theta_1 \frac{n}{\log^{2\alpha+6} n}$, it is sufficient that $k \le \theta_3 \frac{n}{\log^2 n}$ for large n. Then, both Lemmas 11 and 15 are applicable. We assign the two phases the same amount of time, and thus the achievable per-flow date rate is $\min(R_1, R_2)/2 \ge (\frac{1}{2})\min\{c_4, c_7\}\frac{\sqrt{n}}{n_s\sqrt{k}} = c_8\frac{\sqrt{n}}{n_s\sqrt{k}}$. ■

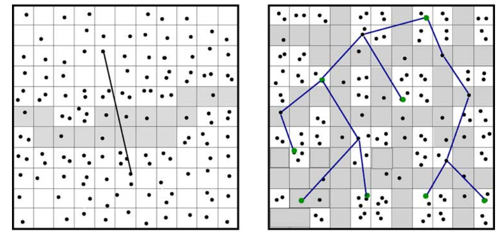
V. Upper Bound on Asymptotic Capacity

In [17], the authors presented an upper bound on the unicast capacity under Gaussian channel model. In [12], an upper bound on multicast capacity under Gaussian channel was presented by use of some novel concepts. Unfortunately, its bounds have discrepancies, *e.g.*, its upper bound on a special case of broadcast ($k = n-1$) is actually smaller than the achievable broadcast capacity known in the literature [29]. In this section, we give a new upper bound for multicast capacity under the Gaussian channel model. The basic idea of our approach is to bound the capacity: 1) studying the largest load of some cell for any routing and scheduling method; and 2) using the capacity bottleneck imposed by some critical link in the network. To study the load of a cell, our method is as follows:

1) First, we partition the region $\mathcal{B}_n = [0, \sqrt{n}] \times [0, \sqrt{n}]$ into cells with a constant side length c.
2) We then obtain a grid graph $\mathcal{F}_n$ consisting of $\frac{n}{c^2}$ cells.
3) We will then analyze the maximum load of cells under any routing and scheduling method for multicast. Here, the load of a cell is defined as the number of flows passing through the cell.

We partition the square region $\mathcal{B}_n$ into cells with constant side length c. We obtain a grid graph $\mathcal{C}_n$ consisting of $m^2 = \frac{n}{c^2}$ cells. Each cell is a vertex in $\mathcal{C}_n$, and two vertices form an edge if the corresponding cells share a common side. See Fig. 6(a) for an illustration. We focus on those cells containing only a constant number of nodes and give the following definition.

Definition 2: We say a cell is a *quasi-closed cell* if it contains at most Δ nodes. Here, Δ is some constant. As illustrated in Fig. 6, we call a path of cells *quasi-closed cut* if it contains only quasi-closed cells and crosses from left to right side of $\mathcal{B}_n$. Furthermore, we define the length of a quasi-closed cut as the total number of cells it contains.



(a) (b)

Fig. 6. Grey cells are the quasi-closed cells. A quasi-closed cell contains at most a constant Δ number of nodes. (a) Quasi-closed cut. (b) Quasi-closed net.

According to the results in [3] and lower tail of Chernoff bounds, we can choose c *small* enough such that $\Omega(m)$ quasi-closed cuts can be partitioned into a number of disjoint groups, each with $\lceil \delta \log m \rceil$ disjoint quasi-closed cuts, and each group is contained inside a slab of size $m \times (\kappa \log m - \varepsilon_m)$, for all $\kappa > 0$, δ small enough, and a nonzero small ε_m such that the side length of each slab is an integer. The same is true when we partition the square into vertical slabs with side length $m \times (\kappa \log m - \varepsilon_m)$. Notice that all of the horizontal and vertical stripes together partition $\mathcal{B}_n$ into *super-cells* with side length $c \cdot (\kappa \log m - \varepsilon_m)$.

For any cell $\mathbf{c}$ and any time slot t, let $\mathcal{I}(t, \mathbf{c})$ be the set of links (s_j, v_j), $1 \le j \le q$, that are scheduled concurrently at time t, with sender s_j or receiver v_j inside $\mathbf{c}$. Let w_i be the achievable data rate of link i in this circumstance. For a given cell $\mathbf{c}$, we first bound the total capacity of links in $\mathcal{I}(t, \mathbf{c})$.

Lemma 17: The throughput capacity of all links in $\mathcal{I}(t, \mathbf{c})$ for any cell $\mathbf{c}$ with a constant side length is of order $O(1)$.

Proof: Let l_j be the length of the link (s_j, v_j). We separate the links into two groups. The first group L_1 contains all links with senders in $\mathbf{c}$, and the second group L_2 contains all links with receivers in $\mathbf{c}$. Let P_j be the transmitting power of sender s_j. Notice that the rate of link (s_j, v_j) is

$$w_j = B\log\left(1+\frac{P_j\cdot\min\{1, l_j^{-\alpha}\}}{N_0+\sum_{k\ne j}P_k\cdot\min\{1, \|s_k-v_j\|^{-\alpha}\}}\right).$$

If we consider only links in L_1, we have, for any link $(s_k, v_k) \in L_1$, $\|s_k - v_j\| \le \|s_k - s_j\| + l_j$. Thus

$$w_j \le B\log\left(1+\frac{P_j\cdot\min\{1, l_j^{-\alpha}\}}{N_0+\min\{1, (l_j+\sqrt{2}c)^{-\alpha}\}\sum_{k\ne j}P_k}\right).$$

Since $c > 0$ is a constant and we assumed that all nodes transmit at the same (or similar) power, it holds that

$$w_j = O\left(\frac{P_j\cdot\min\{1, l_{i(j)}^{-\alpha}\}}{N_0+\min\{1, l_{i(j)}^{-\alpha}\}\sum_{k\ne j}P_k}\right) = O\left(\frac{2P_j}{\sum_{k\in L_1}P_k}\right).$$

Thus, $\sum_{j\in L_1} w_j = O(1)$.

We then consider all links in L_2. In this case, let x be the centroid of the cell $\mathbf{c}$. Let s_1 be the closest sender to x. Then, $\|s_i - v_j\| \le \|s_i - x\| + \|x - v_j\| \le \|s_i - x\| + c/\sqrt{2}$ and $\|s_i - v_j\| \ge \|s_i - x\| - c/\sqrt{2}$. Thus

$$\min\{1, \|s_i-v_j\|^{-\alpha}\} = \Theta(\min\{1, \|s_i-x\|^{-\alpha}\})$$
$$= \Theta(\|s_i-x\|^{-\alpha})$$

when we assume that the sender s_j is out of the cell. Thus, $w_j = O(P_j\|s_j - x\|^{-\alpha}/\sum_{k\in L_2,k\neq j} P_k\|s_k - x\|^{-\alpha})$. For $j \geq 2$, we have

$$\frac{P_j\|s_j - x\|^{-\alpha}}{\sum\limits_{k\in L_2,k\neq j} P_k\|s_k - x\|^{-\alpha}} \leq \frac{2P_j\|s_j - x\|^{-\alpha}}{\sum\limits_{k\in L_2} P_k\|s_k - x\|^{-\alpha}}.$$

For $j = 1$, w_1 is at most a constant. Thus, $\sum_{j\in L_2} w_j = O(1)$.

If all links are considered together, our proof clear still holds. This completes the proof. ∎

For a quasi-closed cell $\mathbf{c}$ and any time slot t, let $\mathcal{X}(t, \mathbf{c})$ be the set of all links that intersect the cell $\mathbf{c}$. Similar to Lemma 17, we can prove the following lemma.

Lemma 18: The throughput capacity of all links in $\mathcal{X}(t, \mathbf{c})$ for any quasi-closed cell $\mathbf{c}$ with a constant side length is $O(1)$.

Proof: Let $\mathbf{c}$ be a quasi-closed cell and x be its centroid. Let $(s_1, v_1), (s_2, v_2), \ldots, (s_g, v_g)$ be the g links that are scheduled concurrently and all intersect the cell $\mathbf{c}$. Let $d_{i,j} = \|s_i - v_j\|$ be the Euclidean distance from s_i to v_j and for simplicity $d_i = d_{i,i}$. It is easy to show that the total capacity achieved by all links with length $d_{i,i} \leq 1$ is at most a constant based on Lemma 18. Then, for simplicity, we assume that $d_{i,i} = \Omega(1)$, for $i \in [1, g]$ and $g > 1$. Then, the total capacity of all links in $\mathcal{X}(t, \mathbf{c})$ is at most (by ignoring all other transmissions)

$$\sum_{i=1}^{g} \log\left(1 + \frac{d_{i,i}^{-\alpha}}{N_0/P + \sum\limits_{j=1,j\neq i}^{g} d_{j,i}^{-\alpha}}\right) < \sum_{i=1}^{g} \log\left(1 + \frac{d_{i,i}^{-\alpha}}{\sum\limits_{j=1,j\neq i}^{g} d_{j,i}^{-\alpha}}\right) < \log e \sum_{i=1}^{g} \frac{d_{i,i}^{-\alpha}}{\sum\limits_{j=1,j\neq i}^{g} d_{j,i}^{-\alpha}}.$$

For any two links (s_i, v_i) and (s_j, v_j) from $\mathcal{X}(t, \mathbf{c})$, let y_i be a point from (s_i, v_i) that is inside the cell and x be the centroid of the cell. It is not difficult to prove that $\|xs_i\| + \|xv_i\| \leq \|xy_i\| + 2\|s_iy_i\| + \|y_iv_i\| \leq d_i + \sqrt{2}c$, where c is the width of cell $\mathbf{c}$. Then, $d_{i,j} + d_{j,i} \leq \|xs_i\| + \|xv_i\| + \|xs_j\| + \|xv_j\| \leq d_i + d_j + 2\sqrt{2}c$. Then, we can show that

$$\frac{\sum\limits_{i=1}^{g} d_{i,i}^{-\alpha}}{\sum\limits_{j=1,j\neq i}^{g} d_{j,i}^{-\alpha}} = O\left(\sum_{i=1}^{g} \frac{d_i^{-\alpha}}{\sum\limits_{j=1,j\neq i}^{g} d_j^{-\alpha}}\right) = O(1).$$

∎

Observe that this lemma does not conflict the arena bound proved in [11] since the arena bound studies the capacity of all links (s_i, v_i) such that the disk $D(s_i, \|s_i - v_i\|)$ contains a given arbitrary point x, while our lemma studies a subset of these links.

We then prove that for any routing method for multicast, there is some cell such that the number of flows whose routing structure will pass through the cell is at least a certain number with high probability. Given a multicast session $\mathcal{M}_k$, let T_k be the multicast tree for $\mathcal{M}_k$ and $C(T_k)$ denote the number of cells passed through by T_k. Here, a cell $\mathbf{c}$ is passed through by a tree T_k if there is a link (s_i, v_i) that intersects the cell $\mathbf{c}$.

Lemma 19: Consider any multicast routing method and a multicast session $\mathcal{M}_k$. We have $C(T_k) = \Omega(\|T_k\|) = \Omega(\sqrt{nk})$.

Proof: For a random multicast session, based on results in [15] and [16], we can show that, *w.h.p.*, the length of any multicast tree T_k for $\mathcal{M}_k$ (with k nodes randomly selected from $\mathcal{B}_n$) is at least $\Omega(\sqrt{k}\sqrt{n})$. Thus, for any routing method for multicast under the Gaussian channel model, *w.h.p.*, the number of cells that will be passed through by a tree T_k will be at least $\lceil \frac{\|T_k\|}{\sqrt{2}c} \rceil = \Omega(\sqrt{nk})$, where c is the side length of a cell $\mathbf{c}$. ∎

We then analyze the maximum load of all quasi-closed cells. Notice that we cannot directly use the total loads of all cells divided by the total number of cells. The reason is that, some routing method may be able to avoid these quasi-closed cells to improve the capacity. Our proof shows that this is impossible by use of super-cells.

Lemma 20: When $n_s = \Theta(n)$, with probability at least $1 - 2e^{-n_s c^2/32}$, the per-session data rate that can be supported using any routing strategy, due to the congestion in some quasi-closed cell, is $O(\frac{1}{n_s} \cdot \frac{\sqrt{n}}{\sqrt{k}})$.

Proof: Recall that a super-cell has side length $\kappa \log m - \varepsilon_m$ and a load of a super-cell under a routing method is defined as the number of flows crossing it. We use L to denote the total load of all super-cells. Note that the number of super-cells crossed by any tree T_k is least $\lceil \frac{\|T_k\|}{\kappa \log m - \varepsilon_m} \rceil$. Obviously, *w.h.p.*, $\|EMST(\mathcal{M}_k)\| = \Omega(m\sqrt{k})$. Similar to Lemma 19, there exists a constant c_1 such that

$$L \geq \sum\nolimits_{i=1}^{n_s} c_1 \cdot \frac{\|EMST(\mathcal{M}_k)\|}{\kappa \log m - \varepsilon_m}.$$

By Azuma's Inequality and Lemma 19, we obtain

$$\Pr\left(L \geq c_3 n_s \sqrt{k} m / \log m\right) \geq 1 - 2e^{-\frac{c_2^2}{32} n_s}$$

for some constants c_2 and c_3. It is not difficult to prove that any multicast routing tree will cross at least $\lceil \delta \log m \rceil$ quasi-closed cuts if it crosses three super-cells. Denoted by $\mathbb{L}'$ the total number of flows crossing some quasi-closed cut. We have $\mathbb{L}' \geq \frac{\mathbb{L}}{3} \times \lceil \delta \log m \rceil$.

It follows that, with probability at least $1 - 2e^{-n_s c_2^2/32}$, the total load of all quasi-closed cells is $\Omega(\sigma(n))$, where $\sigma(n) = (\frac{n_s\sqrt{k}m}{\log m}) \cdot \lceil \delta \log m \rceil$. Then, by pigeonhole principle, with probability at least $1 - 2e^{-n_s c_2^2/32}$, there is at least one quasi-closed cell that will be used by $\Omega(\sigma(n)/m^2)$ flows, which is of order $\Omega(\frac{n_s\sqrt{k}}{\sqrt{n}})$. Then, with probability at least $1 - 2e^{-n_s c_2^2/32}$, the per-session data rate that can be supported using any routing strategy, due to the congestion in some quasi-closed cell, is at most $O(\frac{1}{n_s} \cdot \frac{\sqrt{n}}{\sqrt{k}})$. ∎

Furthermore, we will derive another upper bound based on a result in [22]. That is, for the *random extended network*, the nearest neighbor graph has *w.h.p.*, an edge of length $\Theta(\sqrt{\log n})$. By exploring this long edge, we can derive another upper bound on multicast capacity.

Lemma 21: Under the Gaussian channel model, the per-session multicast capacity for *extended networks* is at most of order $O(\frac{n}{n_s k} (\log n)^{-\frac{\alpha}{2}})$ when $k = \omega(\sqrt{n})$.

Proof: Assume that the longest edge in the nearest neighbor graph of the random network is uv. Then, for node v, the probability $\mathfrak{p}$ that it is chosen as a terminal of a given multicast flow is $\mathfrak{p} = \frac{k}{n}$. It is easy to show that, with probability (at least $1 - e^{-k^2/2n}$), the number of multicast flows that will choose the node v as a terminal is at least $n_s\mathfrak{p}/2$ when $k = \omega(\sqrt{n})$. Observe that the total data rate that node v can receive is at most $R(v) = O((\log n)^{-\frac{\alpha}{2}})$ since the shortest link incident at node v is at least $\Theta(\sqrt{\log n})$. Then, we have the minimum per-session multicast data rate is at most of order $O(\frac{R(v)}{n_s\mathfrak{p}})$, which completes the proof. ■

Combining Lemmas 20 and 21, we get Theorem 2.

VI. Literature Reviews

Gupta and Kumar [7] studied the asymptotic *unicast* capacity of a multihop wireless networks for two different models: random placement and arbitrary placement of nodes. Kulkarni *et al.* [13] obtained a stronger (almost sure) version of the $\sqrt{n \log n}$ throughput for random node locations in a fixed area. Grossglauser and Tse [6] showed that mobility actually can help to improve the unicast capacity if we allow arbitrary large delay. Their main result shows that the average long-term throughput per source–destination pair can be kept constant even as the number of nodes per unit area increases. For random networks, under the protocol model, the achievable per-flow throughput capacity $\lambda(n)$ and the average travel distance $\overline{L}$ satisfies $\lambda(n) \cdot \overline{L} \leq \Theta(\frac{W}{\Delta^2 n \cdot r(n)})$. Similar phenomenon has also been observed in [14]. Gastpar and Vetterli [5] study the capacity of random networks using relay. Chuah *et al.* [2] studied the capacity scaling in MIMO wireless systems under correlated fading. Vu *et al.* [25] studied the scaling laws of cognitive networks. Liu *et al.* [19] studied the capacity of a wireless ad hoc network with infrastructure. Another stream of work (*e.g.,* [21]) has proposed progressively refined multiuser cooperative schemes, which have been shown to significantly outperform multihop communication in many environments. Bounds for the capacity of wireless multihop networks imposed by topology and demand were studied in [11]. Their techniques can be used to study unicast, broadcast, and multicast capacity. Bhandari and Vaidya [1] studied the unicast capacity of multichannel wireless networks with random (c, f) assignment. Garetto *et al.* [4] studied the capacity scaling in delay-tolerant networks with heterogeneous mobile devices. Franceschetti *et al.* [3] show that a per-flow unicast rate $1/\sqrt{n}$ is achievable in networks of randomly located nodes when Gaussian channel is used.

Broadcast capacity of an arbitrary network has been studied in [9] and [24]. They show that, under fPrIM, the broadcast capacity per flow in any network is only $\Theta(W/n)$ if $\Theta(n)$ nodes will serve as sources. This bound also applies to random networks. Keshavarz-Haddad *et al.* [10] studied the broadcast capacity with dynamic power adjustment for physical interference model (PhIM). Zheng [29] studied the data dissemination capacity in power-constrained networks: *w.h.p.*, the total broadcast capacity is $P \cdot \Theta((\log n)^{-\alpha/2})$ when each node transmits at a power P in the Gaussian channel model. Li *et al.* [18] studied the broadcast capacity under PhIM model.

Multicast capacity was also recently studied in the literature. Jacquet and Rodolakis [8] studied the scaling properties of multicast for random wireless networks. They briefly claimed that the maximum rate at which a node can transmit multicast data is $O(\frac{W}{\sqrt{kn \log n}})$. Recently, rigorous proofs of the multicast capacity were given in [15] and [23]. Li *et al.* [15] studied the multicast capacity of the following random networks: n wireless nodes are randomly deployed in a square region with side length a, and each wireless node can transmit/receive at W bits/s over a *common* wireless channel. They proved that, in fPrIM, the per-flow multicast capacity (of n multicast flows, each flow with k receivers) is $\Theta(\sqrt{\frac{1}{n \log n}} \cdot \frac{W}{\sqrt{k}})$ when $k = O(\frac{n}{\log n})$, and is $\Theta(W/n)$ when $k = \Omega(\frac{n}{\log n})$. Shakkottai *et al.* [23] studied the multicast capacity of random networks when the number of multicast sources is n^ϵ for some $\epsilon > 0$ and the number of receivers per multicast flow is $n^{1-\epsilon}$. Recently, Mao *et al.* [20] studied the multicast capacity for hybrid networks under fPrIM model. Wang *et al.* [26] studied the multicast capacity under Gaussian model and show that the per-flow bound $\Omega(\frac{\sqrt{n}}{n_s\sqrt{k}})$ still applies when $k = O(\frac{n}{\log^{\alpha+1} n})$. Wang *et al.* [27] studied capacity scaling laws under (n, m, k)-cast formulation, where n, m, and k denote the number of nodes in the network, the number of destinations for each communication group, and the actual number of communication group members that receive information (i.e., $k \leq m \leq n$), respectively, and when nodes are endowed with multipacket transmission (MPT) or multipacket reception (MPR) capabilities. These results [6]–[10], [15], [23], [24] for the network capacity of random networks all assumed that the data rate supported by each communication link is a constant W bits/s (using PrIM, fPrIM, or PhIM models).

Keshavarz-Haddad and Riedi [12] studied the multicast capacity of large-scale random networks under a variety number of interference models: PrIM, fPrIM, and Gaussian channel model. They proposed some novel concepts: arena and some large separated cluster. They also present a novel constructive lower bound on multicast capacity by partitioning the deployment region using super-cells (with side length $\Theta(\log n)$), large cells (with side length $\Theta(\sqrt{\log n})$), and cells (with side length $\Theta(1)$) for three different purposes. The proofs on the capacity achievable by their routing and scheduling mechanisms are mainly based on the *expected* valuation, which could be far different from the result that needs to be true with high probability. We found that their results have discrepancies when $k > n/\log n$: Their results on total capacity $\Theta(W)$ cannot be achieved by broadcast when $k = n - 1$ [29].

VII. Conclusion

A number of interesting questions remain open. The first question is to derive tight upper bound and lower bound on the network capacity when k could be any arbitrary value from 2 to n. The lower bounds presented here only hold when $k = O(\frac{n}{(\log n)^{2\alpha+6}})$. The second question is to study the capacity when the receiving terminals in a multicast group are within a certain region (e.g., a disk with a radius b or a square with a side length b). Finally, we point out that the problem of optimizing the multicast throughput of a given arbitrary network by choosing best routing protocol and optimizing the hidden constant in our formulas remains open.

TABLE I
NOTATIONS AND ABBREVIATIONS USED IN THIS PAPER

PrIM	Protocol interference model
fPrIM	fixed power (range) protocol interference model
PhIM	Physical interference model
GCM	Gaussian channel model
$\lambda_{k,\mathcal{S}}(n)$	minimum per-flow multicast data rate with sessions $\mathcal{S}$.
$\ell(x)$	path loss for a transmission over a distance x
$R(v_i, \mathcal{D})$	the rate node v_i can send to a set of receivers $\mathcal{D}$ without relay
$D(x, r)$	a disk centered at a point x with radius r
Π_i, Φ_i	a horizontal (vertical) highway produced in the highway system.
$\pi_{i,j}$	the jth node in the ith horizontal highway Π_i
$\phi_{i,j}$	the jth node in the ith vertical highway Φ_i
$X(p)$, $Y(p)$	the x-coordinate and y-coordinate of a point p.
$\mathcal{B}_n$	a 2-dimensional square with side-length $\sqrt{n}$.
$p_{i,x}$	the xth point randomly chosen for the ith multicast.
P_i	the set of points $p_{i,x}$ randomly selected
$v_{i,x}$	the xth node (that is nearest to $p_{i,x}$) in the ith multicast session.
$q_{i,x}$	the node in the highway that is nearest to $v_{i,x}$.
$EST(P_i)$	a Euclidean spanning tree for P_i
$D(T)$	the area covered by transmission disks in a multicast tree T
$d_H(u,v)$, $d_V(u,v)$	the horizontal (vertical) span of the segments connecting u and v in the highway.
$\mathcal{I}(t, \mathbf{c})$	the set of links with an end node inside the cell $\mathbf{c}$ at time-slot t.
$\mathcal{X}(t, \mathbf{c})$	the set of links that intersect the cell $\mathbf{c}$ at time-slot t.
n	the expected number of nodes in the system
n_s	number of multicast sessions.
m	the number of cells per row, *i.e.*, $m = \sqrt{n}/(c\sqrt{2})$.
H	the height (width) of a horizontal (vertical) rectangle produced in deriving highways
B	the bandwidth
P	common transmission power used by all nodes
N_0	variance of background noise
α	path loss exponent

APPENDIX

A. Percolation Theory Result [3]

Consider a square lattice B_m with side length m. We declare each edge of the square grid *open* with probability p, and *closed* otherwise, independently of all other edges.

For any given $\kappa > 0$, let us partition B_m into rectangles R_m^i of sides $m \times (\kappa \log m - \epsilon_m)$. We choose $\epsilon_m > 0$ as the smallest value such that the number of rectangles $\frac{m}{(\kappa \log m - \epsilon_m)}$ in the partition is an integer. It is easy to see that $\epsilon_m = o(1)$ as $m \to \infty$. We let C_m^i be the maximal number of edge-disjoint left-to-right crossings of rectangle R_m^i and let $N_m = \min_i C_m^i$. The result is the following.

Theorem 22 [3]: For all $\kappa > 0$ and $\frac{5}{6} < p < 1$ satisfying $2 + \kappa \log(6(1-p)) < 0$, there exists a $\delta(\kappa, p) > 0$ such that

$$\lim_{m\to\infty} P_p(N_m \le \delta \log m) = 0.$$

B. Chernoff Bound and VC Theorem

Lemma 23: Let X be a Poisson random variable of rate λ.

$$\Pr(X \ge x) \le \frac{e^{-\lambda}(e\lambda)^x}{x^x} \quad \text{for } x > \lambda. \tag{3}$$

Let $\mathcal{U}$ be the input space. Let $\mathcal{C}$ be a family of subsets of $\mathcal{U}$. A *finite* set S (called sample in machine learning) is *shattered* by $\mathcal{C}$, if for *every* subset B of S, there *exists* a set $A \in \mathcal{C}$ such that $A \bigcap S = B$. The *VC dimension* of $\mathcal{C}$, denoted by $\text{VC-d}(\mathcal{C})$, is defined as the maximum value d such that there exists a set S with cardinality d that can be shattered by $\mathcal{C}$. For sets of *finite* VC dimension, one has uniform convergence in the weak law of large numbers:

Theorem 24 (The Vapnik–Chervonenkis Theorem): If $\mathcal{C}$ is a set of *finite* VC dimension $\text{VC-d}(\mathcal{C})$, and $\{X_i \mid i = 1, 2 \ldots, N\}$ is a sequence of *i.i.d.* random variables with *common* probability distribution P, then for every $\epsilon, \delta > 0$

$$\mathbf{Pr}\left(\sup_{A\in\mathcal{C}}\left|\frac{\sum_{i=1}^{N} I(X_i \in A)}{N} - \Pr(A)\right| \le \epsilon\right) > 1 - \delta$$

whenever $N > \max\left\{\frac{8\cdot \text{VC-d}(\mathcal{C})}{\epsilon}\cdot \log\frac{13}{\epsilon}, \frac{4}{\epsilon}\log\frac{2}{\delta}\right\}$.

Here, $I(X_i \in A)$ takes value 1 if $X_i \in A$, and 0 otherwise.

C. Notations and Abbreviations

See Table I.

D. Proof of Some Lemmas

For a region $\mathbb{R}$, and g with $0 \le g \le t-1$, we first run Algorithm 2 line by line. When we run to line 5 for the $(t-g)$th time, for any region $\mathbb{R}$, let $E(\mathbb{R}, g)$ be the event that there is a node from $\mathcal{P}$ that falls in region $\mathbb{R}$. Recall that here $\mathcal{P}$ is the set of nodes representing all connected components (each node for one connected component). We use $\mathbb{D}(p)$ to denote a small enough region that contains point p, and $D(p) = |\mathbb{D}(p)|$ is the area of $\mathbb{D}(p)$. Then, we have the following lemma.

Lemma 25: For any point p in $\mathcal{B}_n$ and $0 \le g \le t$, we have $\Pr\{E(\mathbb{D}(p), g)\} \le (\frac{4^{g+1}}{a^2})D(p)$.

REFERENCES

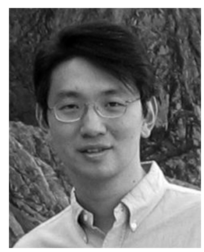
[1] V. Bhandari and N. H. Vaidya, "Capacity of multi-channel wireless networks with random (c, f) assignment," in *Proc. ACM MobiHoc*, 2007, pp. 229–238.
[2] C.-N. Chuah, D. N. C. Tse, J. M. Kahn, and R. A. Valenzuela, "Capacity scaling in MIMO wireless systems under correlated fading," *IEEE Trans. Inf. Theory*, vol. 48, no. 3, pp. 637–650, Mar. 2002.
[3] M. Franceschetti, O. Dousse, D. N. C. Tse, and P. Thiran, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. Inf. Theory*, vol. 53, no. 3, pp. 1009–1018, Mar. 2007.
[4] M. Garetto, P. Giaccone, and E. Leonardi, "Capacity scaling in delay tolerant networks with heterogeneous mobile nodes," in *Proc. ACM MobiHoc*, 2007, pp. 41–50.
[5] M. Gastpar and M. Vetterli, "On the capacity of wireless networks: The relay case," in *Proc. IEEE INFOCOM*, 2002, vol. 3, pp. 1577–1586.
[6] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.
[7] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 1999.
[8] P. Jacquet and G. Rodolakis, "Multicast scaling properties in massively dense ad hoc networks," in *Proc. IEEE ICDCS*, 2005, pp. 93–99.
[9] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi, "Broadcast capacity in multihop wireless networks," in *Proc. ACM MobiCom*, 2006, pp. 239–250.

[10] A. Keshavarz-Haddad and R. Riedi, "On the broadcast capacity of multihop wireless networks: Interplay of power, density and interference," in *Proc. 4th IEEE SECON*, 2007, pp. 314–323.
[11] A. Keshavarz-Haddad and R. H. Riedi, "Bounds for the capacity of wireless multihop networks imposed by topology and demand," in *Proc. ACM MobiHoc*, 2007, pp. 256–265.
[12] A. Keshavarz-Haddad and R. Rudolf, "Multicast capacity of large homogeneous multihop wireless networks," in *Proc. 6th Int. Symp. Modeling Opt. Mobile, Ad Hoc, Wireless Netw.*, 2008, pp. 116–124.
[13] S. Kulkarni and P. Viswanath, "A deterministic approach to throughput scaling in wireless networks," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1041–1049, Jun. 2004.
[14] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. ACM MobiCom*, 2001, pp. 61–69.
[15] X.-Y. Li, S.-J. Tang, and F. Ophir, "Multicast capacity for large scale wireless ad hoc networks," in *Proc. ACM MobiCom*, 2007, pp. 266–277.
[16] X.-Y. Li, "Multicast capacity of wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 950–961, Jun. 2009.
[17] S. Li, Y.-H. Liu, and X.-Y. Li, "Capacity of large scale wireless networks under Gaussian channel model," in *Proc. ACM MobiCom*, 2008, pp. 140–151.
[18] X.-Y. Li, J.-Z. Zhao, Y.-W. Wu, S.-J. Tang, X.-H. Xu, and M. Xu-Fei, "Broadcast capacity for wireless ad hoc networks," in *Proc. IEEE MASS*, 2008, pp. 114–123.
[19] B. Liu, P. Thiran, and D. Towsley, "Capacity of a wireless ad hoc network with infrastructure," in *Proc. ACM MobiHoc*, 2007, pp. 239–246.
[20] X. Mao, X.-Y. Li, and S.-J. Tang, "Multicast capacity for hybrid wireless networks," in *Proc. ACM MobiHoc*, 2008, pp. 189–198.
[21] A. ÖzgÜr, O. LÉvÊque, and D. Tse, "Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3549–3572, Oct. 2007.
[22] M. Penrose, "The random minimal spanning tree in high dimensions," *Ann. Probab.*, vol. 24, pp. 1903–1925, 1996.
[23] S. Shakkottai, X. Liu, and R. Srikant, "The multicast capacity of large multihop wireless networks," in *Proc. ACM MobiHoc*, 2007, pp. 247–255.
[24] B. Tavli, "Broadcast capacity of wireless networks," *IEEE Commun. Lett.*, vol. 10, no. 2, pp. 68–69, Feb. 2006.
[25] M. Vu, N. Devroye, M. Sharif, and V. Tarokh, "Scaling laws of cognitive networks," in *Proc. Int. Conf. Cognitive Radio Oriented Wireless Netw. Commun.*, 2007, pp. 2–8.
[26] C. Wang, X.-Y. Li, C. Jiang, S.-J. Tang, Y.-H. Liu, and J.-Z. Zhao, "Scaling laws on multicast capacity of large scale wireless networks," in *Proc. IEEE INFOCOM*, 2009, pp. 1863–1871.
[27] Z. Wang, H. Sadjadpour, and J. J. Garcia-Luna-Aceves, "Capacity scaling laws of information dissemination modalities in wireless ad hoc networks," in *Proc. 46th Annu. Allerton Conf. Commun., Control Comput.*, 2008, pp. 23–26.
[28] L. Xie and P. Kumar, "A network information theory for wireless communication: Scaling laws and optimal operation," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 748–767, May 2004.
[29] R. Zheng, "Information dissemination in power-constrained wireless networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–10.

Dr. Xiang-Yang Li (SM'08) received the B.Eng. degree in computer science and the Bachelor's degree in business management from Tsinghua University, Beijing, China, in 1995, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 2000 and 2001, respectively.

He has been an Associate Professor since 2006 and was previously an Assistant Professor of computer science at the Illinois Institute of Technology, Chicago, from 2000 to 2006. He was a Visiting Professor with Microsoft Research Asia from May 2007 to August 2008. His research interests span wireless ad hoc and sensor networks, noncooperative computing, computational geometry, and algorithms.

Dr. Li is an Editor of *Networks* and *Ad Hoc & Sensor Wireless Networks: An International Journal*. He was a Guest Editor of special issues for *ACM Mobile Networks and Applications* and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and a Guest Editor of several other journals.

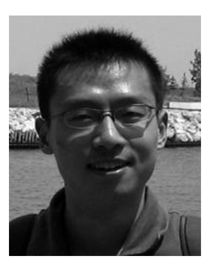


Dr. Yunhao Liu (SM'06) received the B.S. degree in automation from Tsinghua University, Beijing, China, in 1995; the M.A. degree from the Beijing Foreign Studies University, Beijing, China, in 1997; and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, East Lansing, in 2003 and 2004, respectively.

He is now an Associate Professor and the Post-Graduate Director of the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. His research interests include wireless sensor networks, peer-to-peer computing, and pervasive computing.

Shi Li received the B.E. degree in computer science from Tsinghua University, Beijing, China, in 2008.

He is currently a Ph.D. student with the Theory Group of the Computer Science Department, Princeton University, Princeton, NJ. His current research interests include algorithm design and analysis.



ShaoJie Tang received the B.S. degree in radio engineering from Southeast University, Nanjing, China, in 2006.

He has been a Ph.D. student with the Computer Science Department, Illinois Institute of Technology, Chicago, since 2006. His current research interests include algorithm design and analysis for wireless ad hoc networks, wireless sensor networks, and online social networks.