Capacity of Large Scale Wireless Networks Under Gaussian Channel Model*

Shi Li
Dept. of Comp. Sci. & Tech.
Tsinghua University
BeiJing, China

Yunhao Liu
Dept. of Comp. Sci. & Eng.
HK Univ. of Sci. & Tech.
HongKong, China
liu@cse.ust.hk

Xiang-Yang Li
Microsoft Research Asia, &
Dept. of Computer Science
Illinois Inst. of Tech., Chicago
xli@cs.iit.edu

ABSTRACT

In this paper, we study the multicast capacity of a large scale random wireless network. We simply consider the extended multihop network, where a number of wireless nodes $v_i (1 \leq i \leq n)$ 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 constant power P, and the power decays along path, with attenuation exponent $\alpha > 2$. The data rate of a transmission is determined by the SINR as $B \log(1 + \text{SINR})$. There are n_s randomly and independently chosen multicast sessions. Each multicast 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. Our result generalizes the unicast capacity [3] for random networks using percolation theory.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication, Network topology; G.2.2 [**Graph Theory**]: Network problems, Graph algorithms

General Terms

Algorithms, Design, Theory

Keywords

Wireless ad hoc networks, capacity, multicast, broadcast, unicast, scheduling, optimization, probability theory, percolation theory.

1. INTRODUCTION

In many applications, *e.g.*, wireless sensor networks, we often need an estimation on the (asymptotic) achievable throughput when we randomly deploy 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 when we choose the *best* protocols for all layers. As in the literature, we will mainly consider one type of networks, large scale *random networks*, where a large number of nodes are randomly placed in the deployment region.

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

1) **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 transmitting node $v_k \neq v_i$. Here η is a constant.

2) **Fixed-Power Protocol Interference Model (fPrIM)**: In this model, each node $v \in V$ has a fixed constant transmission range r and a fixed constant interference range $R > r$. 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 such that $\|w - u\| \leq R$ and node w is transmitting simultaneously with node v. Here $\|w - u\|$ is the Euclidean distance between w and u.

3) **Physical Interference Model(PhIM)**: At any time, given a set of simultaneous transmitting nodes $A = \{u_1, u_2, \cdots, u_a\}$, a node v can successfully receive the signal from a sender u iff $\text{SINR} = \frac{P_u \cdot \ell(u,v)}{N_0 + \sum_{i=1}^{a} P_{u_i} \ell(u_i,v)} \geq \sigma$. Here σ is a threshold for SINR, P_{u_i} is the transmission power of node u_i, $\ell(u_i, v)$ is the path loss of signal propagation, and N_0 is the variance of background noise.

4) **Gaussian Channel Model (GIM)**: At any time instance, given a set of simultaneous transmitting nodes $A = \{u_1, u_2, \cdots, u_a\}$. a node v can successfully receive the signal from a sender u_j at a data rate $B \log(1 + \text{SINR})$, where $\text{SINR} = \frac{P_{u_j} \cdot \ell(u_j,v)}{N_0 + \sum_{i=1}^{a} P_{u_i} \ell(u_i,v)}$ and B is the bandwidth of the channel.

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 ground breaking 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 destination located at a nonvanishingly small distance away, the amount of information that can be exchanged by each source-destination pair must go to zero, as $n \to \infty$, 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 (fPrIM or PhIM with assumption $\ell(u,v) = 1/\|u - v\|^{\alpha}$ for a constant $\alpha > 2$) [7]. It has later been extended to hold in a more general information theoretic setting [22]. 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 $1/\sqrt{n \log n}$, by using a specific multihop strategy, when fPrIM or PhIM models are used. For Gaussian channel model, using multihop transmission, pairwise coding and decoding at each hop, and a TDMA scheme, Franceschetti *et al.* [3] shows that a rate $\Omega(1/\sqrt{n})$ is achievable in networks of randomly located nodes (not only some arbitrarily placed nodes). Hence, there is no gap between the capacity of randomly located, and arbitrarily located nodes, at least up to a constant scaling, although using different channel models.

In this paper, we will concentrate on the *multicast capacity* of a random wireless network, which generalizes both the unicast capacity [7] and broadcast capacity [9, 20] for random networks. We assume that a set of wireless nodes $V = \{v_1, v_2, \cdots, v_n, \cdots\}$ are *randomly* distributed (with Poisson distribution of rate 1) in a square region B_n with a side-length $a = \sqrt{n}$ and all nodes transmit at a constant power P. Assume that a subset $S \subseteq V$ of $n_s = |S|$ random nodes will serve as the source nodes of n_s multicast sessions. We randomly and independently choose n_s multicast sessions as follows. To generate the i-th ($1 \le i \le n_s$) multicast session, k points $p_{i,j} (1 \le j \le k)$ are randomly and independently chosen from the deployment region B_n. Let $v_{i,j}$ be the nearest wireless node from $p_{i,j}$ (ties are broken randomly). In the i-th multicast session, $v_{i,1}$ will multicast data to $k - 1$ nodes $U_i = \{v_{i,j} \mid 2 \le j \le k\}$ at an arbitrary data rate λ_i. The aggregated multicast capacity with $S = \{v_{1,1}, v_{2,1}, \cdots, v_{n_s,1}\}$ as roots for a network is defined as $\Lambda_{k,S}(n) = \sum_{v_i \in S} \lambda_i$ when there is a schedule of transmissions such that all multicast flows will be received by their destination nodes successfully within a finite delay. Similarly, we define the minimum per-flow multicast throughput (or capacity) as $\lambda_{k,S}(n) = \min_{v_i \in S} \lambda_i$. Our result will show how the multicast capacity of wireless networks scale with the number of nodes in the networks, or scale with the size of the deployment region, or scale with the size of multicast group.

Using fixed-power protocol interference model fPrIM, Li *et al.* [14] and Shakkottai *et al.* [19] showed that, when there are n_s multicast flows and each multicast flow will have k receivers, the perflow multicast capacity of n_s flows for random networks is of order $\frac{W\sqrt{n}}{n_s\sqrt{k \log n}}$ when $k = O(n/\log n)$, and is of order W/n_s when $k = \Omega(n/\log n)$. Although protocol interference model can approximate the interference to some extent, experiment studies show that they are still much different from the practice. In this paper, we study the asymptotic network capacity using the Gaussian Channel model. 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 multi-hop manner until they reach their final destinations. The packets could be buffered at intermediate nodes while awaiting 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 future work to study the scenario when network coding is permitted, the buffers of intermediate nodes are

bounded by some values. In some results, we assume that every intermediate node have infinite buffer size. For most of the results presented here, the delay of the routing is not considered, *i.e.*, the delay in the worst case could be arbitrarily large for some results.

Our Main Contributions: This paper shows that a per-flow multicast rate $1/\sqrt{nk}$ is achievable in networks of n randomly located nodes in a square region $B_n = \sqrt{n} \times \sqrt{n}$. Specifically, we will prove the following main theorem.

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

$$c_8 \frac{\sqrt{n}}{n_s\sqrt{k}}$$

where constants $c_8 = \frac{1}{2}\min\{c_4, c_7\}$, $c_4 = \frac{2^{\alpha}BP}{c_0 N_0 \sqrt{\theta_1}}$, and $c_7 = \frac{B}{100((97\kappa\sqrt{\theta_3}+24)h+1)}\log\left(1 + \frac{P\cdot(2\sqrt{2}c)^{-\alpha}}{N_0+22P(2\sqrt{2}c)^{-\alpha}}\right)$. Here c_0, c, κ are constants.

In terms of capacity upper bound, we proved that

THEOREM 2. *The asymptotic per-flow unicast capacity of n flows in a large scale random network with n nodes randomly distributed in a square B_n is at most of order $1/\sqrt{n}$.*

Compared with [14, 19], 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 perflow unicast capacity proved in Theorem 2 shows that the capacity achieved by [3] is asymptotically optimal, and thus closes the gap when Gaussian link model is used.

The rest of the paper is organized as follows. In Section 2, we briefly describe the network and system model used throughout the paper. Our routing strategy that can achieve asymptotic optimal multicast capacity is presented in Section 3. We present the theoretic analysis in Section 4 and present a matching upper bound for asymptotic per-flow unicast capacity in Section 5. We review the related work in Section 6 and conclude the paper in Section 7.

2. NETWORK AND SYSTEM MODEL

Consider a square region 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 1. Assume that each node can transmit at 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 $\ell(d)$ is the transmission loss during a path of length d. In this paper, we consider the attenuation function

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

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

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

---

[1] Here an event is said to happen with high probability (*w.h.p.*), if for any $0 < \epsilon < 1$, there is a large integer N (typically $N = 1/\epsilon$) such that for any random network of size at least N, the probability that the event happens is at least $1 - \epsilon$.

where $\mathcal{A}$ is the set of nodes transmitting simultaneously with node v_i, B is the channel bandwidth, N_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).

We choose n_s nodes to be the sources of the multicast sessions. For each source node, 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

1: **for** $i \leftarrow 1, 2, \cdots, n_s$ **do**
2:     **for** $j \leftarrow 1, 2, \cdots, k$ **do**
3:         Randomly choose a point $p_{i,j}$ in B_n.
4:         Choose a node $v_{i,j}$ from V that is closest to $p_{i,j}$
5:     **end for**
6:     Let $v_{i,1}$ be a source node and $v_{i,2}, v_{i,3}, \cdots, v_{i,k}$ be its intended receivers.
7: **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 be also 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 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 can not combine the receivers of these two multicast sessions together either.

Given a random wireless network of n nodes and the set $\mathcal{S}$ of $n_s = |\mathcal{S}|$ source nodes, let $\lambda_{\mathcal{S}} = (\lambda_{i_1}, \lambda_{i_2}, \cdots, \lambda_{i_{n_s-1}}, \lambda_{i_{n_s}})$ 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 $1 \le j \le n_s$. When given a *fixed* network $G = (V, E)$, where the node positions of all nodes V, 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/sec is *feasible* if there is a spatial and temporal scheme for scheduling transmissions such that by operating the network in a multi-hop fashion and buffering at intermediate nodes when awaiting transmission, every node v_i can send λ_i bits/sec 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.

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 $\lambda_{k,\mathcal{S}}^a(n) = \frac{\sum_{v_i \in \mathcal{S}} \lambda_i}{n_s}$. 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/sec is *feasible* for n_s multicast sessions (each session with k terminals) if there is a rate vector $\lambda_{\mathcal{S}} = (\lambda_{i_1}, \lambda_{i_2}, \cdots, \lambda_{i_{n_s-1}}, \lambda_{n_s})$ 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.

DEFINITION 1   (CAPACITY OF RANDOM NETWORKS). *We say that the* multicast capacity per flow *of a class of random networks is of order $\Theta(f(n))$ bits/sec if there are deterministic constants $c > 0$ and $c < c' < +\infty$ such that*

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

Here the probability is computed using *all* possible connected random networks formed by n nodes distributed in a square with side-length a. We will study the per-flow multicast capacity under Gaussian channel model, instead of the fPrIM used in [14, 19].

3. OUR 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]; we then present our multicast method based on the highway system; we finally analyze the performance of our multicast method.

3.1 Technical Lemmas

To study the asymptotic multicast capacity, we first present some technical lemmas that are essential for the analysis.

LEMMA 3. *At any time instance, assume that for* any *receiver v_i, the following two conditions are satisfied:*
- *C_1: v_i is within Euclidean distance r from its sender v_j; and*
- *C_2: for any other sender $v_k (k \ne j)$, the Euclidean distance between v_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.

PROOF. Let V_S be the set of senders (which have at least one intended receiver), and V_R be the set of receivers. So, $V_S \bigcap V_R = \emptyset$. If conditions C_1 and C_2 are satisfied, any two senders are at least $R' = R - r$ away from each other. For any receiver $v^* \in V_R$ and non-negative integer g, let

$$\mathcal{N}_g(v^*) = \left\{ v \in V_S \mid gR' \le d(v^*, v) < (g+1)R' \right\}.$$

Let v' be the intended sender of v^*, and $n_g(v^*) = |\mathcal{N}_g(v^*)|$ be the size of $\mathcal{N}_g(v^*)$. If we divide the ring (centered at node v^*) into $t_g = 2\lceil \frac{\pi(g+1)R'}{R'/2} \rceil = 2\lceil \pi(2g+2) \rceil$ sectors (see Figure 1), the distance of any two points in the same sector is at most R'. Here the ring is divided as follows: We first divide the ring into $\frac{\pi(g+1)R'}{R'/2}$ sectors, then each sector is divided into two sectors by a circle with radius $(g + 1/2)R'$. Thus, a sector contains at most 1 sender, i.e.,

$$n_g(v^*) \le t_g = 2\lceil \pi(2g+2) \rceil.$$

Since $n_0(v^*) = 0$, the total signal interference at node v^* by all other transmitting nodes is $I(v', v^*) \le \sum_{g=1}^{\infty} n_g(v^*) P \cdot \ell(gR') \le \sum_{g=1}^{\infty} 2\lceil \pi(2g+2) \rceil P(gR')^{-\alpha} \le PR'^{-\alpha} \sum_{g=1}^{\infty} 2\lceil \pi(2g+2) \rceil g^{-\alpha}$. Obviously, the sum in the rightmost inequality converges if $\alpha > 2$. So, $I(v', v^*) \le c_1 P(R - r)^{-\alpha}$, where c_1 is a constant. Thus,

$$
\begin{aligned}
R(v', v^*) &= B \log \left(1 + \frac{S(v', v^*)}{N_0 + I(v', v^*)} \right) \\
&\ge B \log \left(1 + \frac{P \cdot \ell(r)}{N_0 + c_1 P(R - r)^{-\alpha}} \right),
\end{aligned}
$$

where $c_1 = \sum_{g=1}^{\infty} 2\lceil \pi(2g+2) \rceil g^{-\alpha}$ is a constant if $\alpha > 2$.   □

LEMMA 4. *For $\gamma > 0$, if we partition the square $B_n = [0, \sqrt{n}] \times [0, \sqrt{n}]$ into at least $\tau_1 \frac{n}{\log^\gamma n}$ subsquare regions 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. (τ_1 and τ_2 are constants.)*

PROOF. Let A_n be the event that there are more than $2\tau_2 \log^\gamma n$ nodes in some subsquare. Then by the union bound and Chernoff bound (Lemma 24), the probability of event A_n is

$$
\begin{aligned}
\Pr(A_n) &\leq \left\lceil \tau_1 \frac{n}{\log^\gamma n} \right\rceil \frac{e^{-\tau_2 \log^\gamma n}(e\tau_2 \log^\gamma n)^{2\tau_2 \log^\gamma n}}{(2\tau_2 \log^\gamma n)^{2\tau_2 \log^\gamma n}} \\
&= \left\lceil \tau_1 \frac{n}{\log^\gamma n} \right\rceil e^{-\tau_2 \log^\gamma n} \left(\frac{e}{2} \right)^{2\tau_2 \log^\gamma n} \\
&= \left\lceil \tau_1 \frac{n}{\log^\gamma n} \right\rceil \left(\frac{e}{4} \right)^{\tau_2 \log^\gamma n} \to 0
\end{aligned}
$$

as n tends to infinity. $\square$

Observe that when $\gamma > 1$, $\Pr(A_n) < \frac{\tau_1}{n^{\tau_2 \log(4/e)-1}}$.

LEMMA 5. *If we partition B_n into regions of area at least $a \log n$ (for $a \geq 1$), then w.h.p every region contains at least 1 node.*

PROOF. Let A_n be the event that some region is empty of nodes. Then $\Pr(A_n) \leq \left\lceil \frac{n}{a \log n} \right\rceil \left(1 - \frac{a \log n}{n} \right)^n \leq \left\lceil \frac{n}{a \log n} \right\rceil e^{-a \log n} = \left\lceil \frac{n}{a \log n} \right\rceil \frac{1}{n^a} \to 0$ as n tends to infinity. That's to say, w.h.p, there are at least 1 node in every region. $\square$

3.2 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 highway system, we partition the deployment box B_n into subsquares s_i of a constant side length c, as depicted in Figure 2. In Figure 2, let $X(s_i)$ be the number of random nodes inside 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 $p \equiv \Pr(X(s_i) \geq 1) = 1 - e^{-c^2}$. 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. Observe that the event whether a square s_i is open is *independent* of the event whether another square s_j is open. Thus, percolation theory can be applied here. This model is then mapped into a discrete edge-percolation model on the square grid.

We associate an edge to each square, traversing it diagonally, as depicted on the right-hand side of the Figure 2. The edge is said to be either open or closed according to the state of the corresponding square. 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 edge implies that we have a routing path such that the data rate achievable by this path is of a constant value (depending on c) from Lemma 3, using a TDMA scheduling of nodes. Note that, when constant c is large enough, the preceding construction produces winding open paths that cross the entire network area. Denote the number of edges composing the side length of B_n by $m = \frac{\sqrt{n}}{c\sqrt{2}}$, where c is rounded up such that m is an integer. By Theorem 23, we can choose c large enough such that, *w.h.p.*, there are $\Omega(m)$ paths crossing B_n from left to right. These paths can be grouped into disjoint sets of paths: each group have $\lceil \delta \log m \rceil$ paths, crossing a rectangle of width m and height $\kappa \log m - \epsilon_m$, for all $\kappa > 0$, δ small enough, and a vanishing

small ϵ_m so that the side length of each rectangle is an integer. See Figure 3 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, that was called the *highway system* [3].

We then slice each horizontal rectangle into horizontal strips of constant height 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 $\Pi_1, \Pi_2, \cdots, \Pi_M$ 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 i-th rectangle. We also let $\pi_{i,j}$ be the j-th node in the i-th horizontal highway. So, a highway Π_i can be denoted by a list of nodes, i.e, $\Pi_i = (\pi_{i,1}, \pi_{i,2}, \cdots, \pi_{i,s_i})$. Similarly, we use $\Phi_1, \Phi_2, \cdots, \Phi_M$ to denote the M vertical highways, where $\Phi_i = (\phi_{i,1}, \phi_{i,2}, \cdots, \phi_{i,t_i})$. In this paper, we propose the following definition that will be used in our proofs later.

DEFINITION 2. *We call a horizontal(vertical) highway $\Pi_i = (\pi_{i,1}, \pi_{i,2}, \cdots, \pi_{i,s_i})(or \Phi_i = (\phi_{i,1}, \phi_{i,2}, \cdots, \phi_{i,t_i}))$ legal 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.*

In the appendix, we will prove 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 legal horizontal highways and M legal vertical highways.*

3.3 Schedule the multicast tasks

We now are ready to describe our multicast method. 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* sometime) 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.

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 non-station 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 Figure 4.
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.
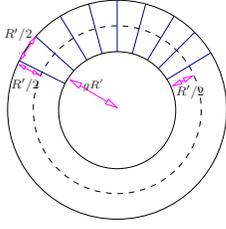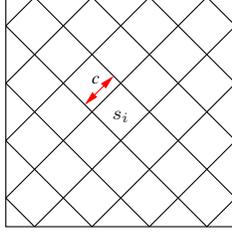
Figure 1: Divide a ring into sectors.

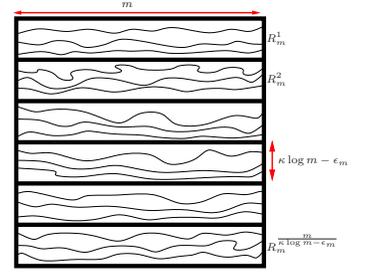**Figure 2: Construction of the bond percolation model.**
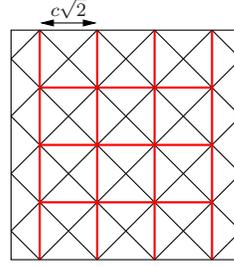

Figure 3: There exists a large number of crossing paths in B_m.
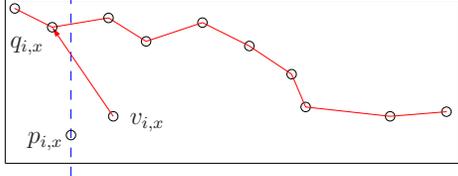

Figure 4: Choose $q_{i,x}$ for $v_{i,x}$ where the path is a highway.

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.
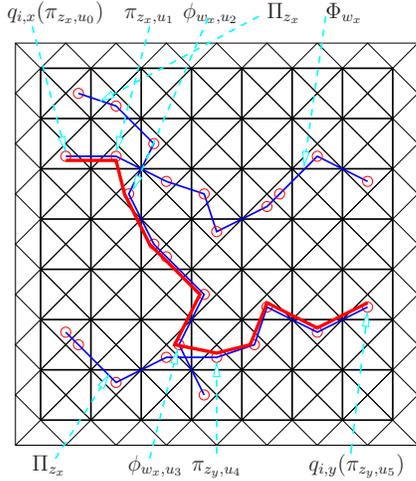
Figure 5: A path connecting $q_{i,x}$ and $q_{i,y}$ contains 3 highway segments: the horizontal one from $q_{i,x}$ to π_{z_x,u_1}, the vertical one from ϕ_{w_x,u_2} to ϕ_{w_x,u_3}, and the horizontal one from π_{z_y,t_4} to $q_{i,y}$. These 3 segments are connected by shortcuts, $\pi_{z_x,u_1}\phi_{w_x,u_2}$ and $\phi_{w_x,u_3}\pi_{z_y,u_4}$, of length at most $\sqrt{5}c$.

We first describe our method (Algorithm 2) to construct an Euclidean spanning tree of a set P_i of k points. We have to point out that our method will not necessarily construct an Euclidean minimum spanning 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 an 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 order of the Euclidean length of Euclidean minimum spanning tree.

Algorithm 2 Find a Euclidean Spanning Tree for k points

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

1: $t \leftarrow$ the minimum integer such that $4^t \geq k$;
2: $\mathcal{P} \leftarrow P_i$
3: $\mathcal{E} \leftarrow \emptyset$;
4: **for** $g \leftarrow t - 1, \cdots, 1, 0$ **do**
5:     Divide B_n into $2^g \times 2^g$ cells, each with size $\frac{a}{2^g} \times \frac{a}{2^g}$;
6:     **for** each cell of size $\frac{a}{2^g} \times \frac{a}{2^g}$ **do**
7:         **if** the cell contains $s \geq 2$ points in $\mathcal{P}$ **then**
8:             Randomly choose a point $p_{i,x}$ from these s points;
9:             **for** any other point $p_{i,y}(y \neq x)$ in this cell **do**
10:                 $\mathcal{E} \leftarrow \mathcal{E} \bigcup \{\overline{p_{i,x}p_{i,y}}\}$
11:                 $\mathcal{P} \leftarrow \mathcal{P} - \{p_{i,y}\}$;
12:             **end for**
13:         **end if**
14:     **end for**
15: **end for**
16: Output $\mathcal{E}$ as the edges of $EST(P_i)$.

---

4. ANALYSIS OF CAPACITY

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

4.1 Data rate of the first phase

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. While in the second step, we calculate how many multicast sessions a non-station node v^* is covered by, which will imply the data rate achievable in 1st and 3rd phase.

LEMMA 7. *In the first (and 3rd) 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 date rate achievable by our method between a terminal $v_{i,x}$ and the highway en-*

Algorithm 3 Build a multicast tree using highway

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

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

1: **for** $x \leftarrow 1, 2, \cdots, k$ **do**
2:     Suppose $p_{i,x}$ is in the z_x-th horizontal strip;
3:     Let $q_{i,x}$ be the node from Π_{z_x} which is closest to the vertical line drawn from $p_{i,x}$(see Figure 4);
        ▷ $q_{i,x}$ will relay data for $v_{i,x}$.
4: **end for**
5: **for** each edge $\overline{p_{i,x}p_{i,y}}$ in $EST(P_i)$ **do**
6:     Suppose $q_{i,x} = \pi_{z_x,u_0}$, and $q_{i,y} = \pi_{z_y,u_5}$;
7:     **if** $z_x = z_y$ **then**
8:         $E(q_{i,x}, q_{i,y}) \leftarrow (\pi_{z_x,u_0}, \pi_{z_x,u_0\pm 1}, \cdots, \pi_{z_x,u_5})$.
9:     **else**
10:         Suppose $p_{i,x}$ is on the w_x-th vertical strip.
11:         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 $d(\pi_{z_x,u_1}, \phi_{w_x,u_2}) \leq \sqrt{5}c$;
12:         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 $d(\phi_{w_x,u_3}, \pi_{z_y,u_4}) \leq \sqrt{5}c$;
13:         $E_1(q_{i,x}, q_{i,y}) \leftarrow (\pi_{z_x,u_0}, \pi_{z_x,u_0\pm 1}, \cdots, \pi_{z_x,u_1})$;
14:         $E_2(q_{i,x}, q_{i,y}) \leftarrow (\phi_{w_x,u_2}, \phi_{w_x,u_2\pm 1}, \cdots, \phi_{w_x,u_3})$;
15:         $E_3(q_{i,x}, q_{i,y}) \leftarrow (\pi_{z_y,u_4}, \pi_{z_y,u_4\pm 1}, \cdots, \pi_{z_y,u_5})$;
16:         $E(q_{i,x}, q_{i,y}) \leftarrow E_1(q_{i,x}, q_{i,y}) \propto E_2(q_{i,x}, q_{i,y}) \propto E_3(q_{i,x}, q_{i,y})$;   ▷ See Figure 5 for illustration, $\propto$ means concatenation of paths.   ▷ Here $E(q_{i,x}, q_{i,y})$ is a path in the highway system connecting $q_{i,x}$ and $q_{i,y}$ (See Figure 5).
17:     **end if**
18: **end for**
19: Let $MT'(V_i)$ be the set of edges that covered by any path $E(q_{i,x}, q_{i,y})$, union the set $\{\overline{q_{i,x}v_{i,x}} \mid 1 \leq x \leq k\}$.
20: $MT'(V_i)$ is a connected graph that covers V_i. We can break the cycles of the graph by removing some edges and the resulted graph would be a tree. Let $MT(V_i)$ be the resulted tree.

---

trance 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 1 node in every region with area $\log n$. Thus, we could divide square 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 $\sqrt{2}(1 + \xi_n)\sqrt{\log n}$, w.h.p..

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 $m = \sqrt{n}/(c\sqrt{2})$.

Then 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 $B \log \left(1 + \frac{P \cdot \ell(r)}{N_0 + c_1 P(R-r)^{-\alpha}}\right)$ when the condition C2 of Lemma 3 is satisfied. This condition can be guaranteed by

---

dividing the phase 1 into time slots. We partition the square B_n into a number of subsquares with length r, and divide the phase 1 into 16 time slots such that within a time slot, any two subsquares that contain transmitting nodes is at least 4 subsquares away (See Figure 6 (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 1 transmitting node at each subsquare, 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 *w.h.p.*, since, *w.h.p.*, each subsquare contains at most $2r^2$ nodes. Considering the number of mini time slots, we could see that w.h.p, the data rate between each pair of $v_{i,x}$ and $q_{i,x}$ that we can achieve is at least

$$B \log \left(1 + \frac{P \cdot \ell(r)}{N_0 + c_1 P(R-r)^{-\alpha}}\right) / (16 \times 2r^2)$$
$$\geq (1 - \varepsilon_1)BP \cdot r^{-\alpha} / (32N_0 r^2) = (1 - \varepsilon_1)\frac{BP}{32N_0}r^{-\alpha-2}$$
$$\geq (1 - \varepsilon_1)\frac{BP}{32N_0}\left((1 + \varepsilon_2)\frac{\log n}{2}\right)^{-\alpha-2}$$
$$= \frac{2^\alpha BP}{16N_0}(\log n)^{-\alpha-2}(1 - \epsilon_1)(1 + \epsilon_2)^{-\alpha-2}$$
$$\geq \frac{2^\alpha BP}{17N_0}(\log n)^{-\alpha-2}$$

The above inequality requires that n is sufficient 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 direction can achieve the lower bound. Setting $c_2 = \frac{2^\alpha BP}{17N_0}$ will finish our proof. □

Figure 6: (a) The subsquares that contain transmitting nodes are at least 4 subsquares away from each other, and each subsquare contains at most 1 transmitting node. In the figure, the nodes with arrows represent transmitting nodes. (b) The subsquares where v^* may be located. p^* is located in the square in the center, and the green squares and the center square (totally 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^*.

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 B_n and let v^* be its nearest wireless node. We then are asking, what is the probability that a node v^* is chosen in this process $\mathcal{Q}$? The following lemma gives the answer.

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 B_n into subsquares of side-length $(1 + \xi_n)\sqrt{\log n}$ and w.h.p each subsquare contains at least 1 node. Considering a point p^* in a subsquare s, *w.h.p.*, its nearest node v^* must fall in s or the 20 subsquares around s (see Figure 6 (b)). To speak in another way, if v^* is in a subsquare s', p^* must fall in s' or the 20 subsquares around s'. So, 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. So, if we let $c_3 = 22$, w.h.p, for any station v^*, the probability is at most $c_3\frac{\log n}{n}$. $\square$

LEMMA 9. *W.h.p, for any non-station node v^*, the probability that a multicast session has v^* as a receiver is at most $c_3 k\frac{\log n}{n}$.*

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 $c_3 k\frac{\log n}{n}$. $\square$

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 $3c_3 n_s k\frac{\log n}{n}$ when $n_s k \geq n$.*

PROOF. Let A_n be the event that a node v^* is chosen by $\mathcal{Q}$ more than $3c_3 n_s k\frac{\log n}{n}$ times. Let $p = c_3 k\frac{\log n}{n}$, the probability that v^* is chosen as terminal of a multicast session. Then

$$\Pr(A_n) \leq n_s \binom{n_s}{3n_s p} p^{3n_s p} \leq n_s \left(\frac{n_s e}{3n_s p}\right)^{3n_s p} p^{3n_s p}$$

$$\leq n_s \left(\frac{n_s e}{3n_s}\right)^{3n_s p} \leq n_s \left(\frac{e}{3}\right)^{3n_s p} \leq n_s \left(n^{-3c_3(\log 3 - 1)}\right)^{\frac{n_s k}{n}}$$

$$\to 0 \text{ (notice that } 3c_3(\log 3 - 1) > 1 \text{ and } n_s k \geq n)$$

This finishes the proof. $\square$

LEMMA 11. *W.h.p, there exist a constant $c_4 > 0$, the data rate that any multicast session can achieve in the first 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 $n_s \geq \theta_2 n^{1/2+\beta}$, 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 Lemma 7 and Lemma 10, w.h.p., the data rate achievable per-multicast session in the first phase is

$$R_1^1 \geq \frac{c_2(\log n)^{-\alpha-2}}{3c_3 n_s k\frac{\log n}{n}} = \frac{c_2}{3c_3}\frac{n(\log n)^{-\alpha-3}}{n_s k}$$

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

$$= \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{\sqrt{n}}{n_s\sqrt{k}}$$

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

$$R_1^2 \geq \frac{c_2(\log n)^{-\alpha-2}}{3c_3 n\frac{\log n}{n}} \geq \frac{c_2}{3c_3}(\log n)^{-\alpha-3}$$

$$\geq \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{n^{-\beta}}{\sqrt{k}} \geq \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{\sqrt{n}}{n_s\sqrt{k}}$$

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 $n_s \geq \theta_2 n^{1/2+\beta}$,

$$R_1 \geq \frac{c_2}{3c_3\sqrt{\theta_1}}\frac{\sqrt{n}}{n_s\sqrt{k}}$$

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

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.

4.2 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. For a region $\mathbb{R}$, and $g(0 \leq g \leq t - 1)$, 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 12. *For any point p in B_n and $0 \leq g \leq t$, we have*

$$\Pr\{E(\mathbb{D}(p), g)\} \leq \frac{4^{g+1}}{a^2}D(p).$$

PROOF. For $g \leq t - 2$, at line (5) of Algorithm 2, there is at most one representative wireless node in each $\frac{a}{2^{g+1}} \times \frac{a}{2^{g+1}}$ cells. Furthermore, we can see if there is a node in a cell s, this node is randomly located in s. i.e, each point in s has the same probability density $\frac{1}{a^2/4^{g+1}} = \frac{4^{g+1}}{a^2}$ to be the node. So, when $g \leq t - 2$, for each point p, $\Pr\{E(\mathbb{D}(p), g)\} \leq \frac{4^{g+1}}{a^2}D(p)$.

When $g = t - 1$, since there are k nodes in $\mathcal{P}$, we have

$$\Pr\{E(\mathbb{D}(p), g)\} \leq \frac{k}{a^2}D(p) \leq \frac{4^t}{a^2}D(p) = \frac{4^{g+1}}{a^2}D(p).$$

So, for $0 \leq g \leq t - 1$, we have $\Pr\{E(\mathbb{D}(p), g)\} \leq \frac{4^{g+1}}{a^2}D(p)$. This finishes the proof. $\square$

LEMMA 13. *For any region $\mathbb{R}$ in B_n and $0 \leq g \leq t - 1$,*

$$\boldsymbol{Pr}\left(E(\mathbb{R}, g)\right) \leq \frac{4^{g+1}}{a^2}|\mathbb{R}|$$

PROOF. The probability is computed by integration: $\Pr\{E(\mathbb{R}, g)\} = \oiint_{p \in \mathbb{R}} \Pr\{E(\mathbb{D}(p), g)\} \leq \oiint_{p \in \mathbb{R}} \frac{4^{g+1}}{a^2}D(p) = \frac{4^{g+1}}{a^2}|\mathbb{R}|$. $\square$

LEMMA 14. *In the second phase, the probability that a station 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.*

PROOF. Considering Algorithm 3, we can see a highway node v^* can be covered by a multicast session in the following two cases.
1. v^* is covered by a horizontal path $E(q_{i,x}, q_{i,y})$ got by line 8 of Algorithm 3.
2. v^* is covered by a horizontal path $E(q_{i,x}, q_{i,y})$ got by Line 16 of Algorithm 3.

We now study these two cases separately.

Case 1: v^* is covered by a horizontal path $E(q_{i,x}, q_{i,y})$ got by line 8 of Algorithm 3.

In this case, $q_{i,x}$ must be in the same horizontal highway with v^*, say, Π_{z_x}. It means that $p_{i,x}$ must be in the z_x-th horizontal strip. Thus, the vertical span of $p_{i,x}$ is h. Consider the value of g
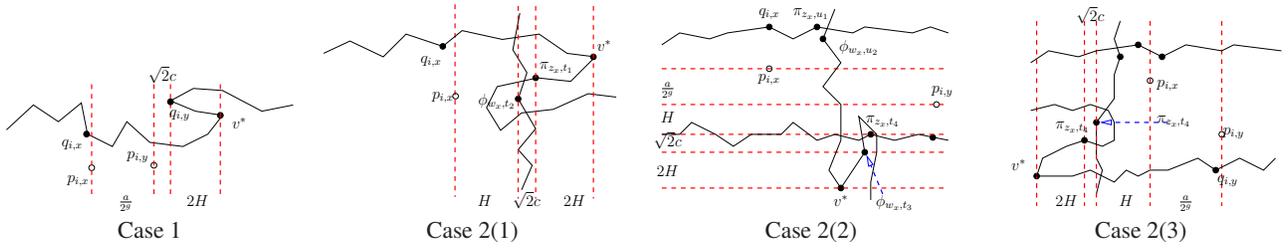
Figure 7: The cases in which the station v^* is covered. In all cases, either $p_{i,x}$ or $p_{i,y}$ is bounded in a rectangle with size at most $h \times \left(\frac{a}{2^g} + 3H + \sqrt{2}c\right)$.

at the line 10 of Algorithm 2 when $\overline{p_{i,x}p_{i,y}}$ is inserted into EST. We can see both the horizontal and the vertical span of $\overline{p_{i,x}p_{i,y}}$, $d_H(p_{i,x}, p_{i,y})$, are at most $\frac{a}{2^g}$. So, we will show the upper bound of the vertical span of $p_{i,x}$ on g. Since v^* is between $q_{i,x}$ and $q_{i,y}$ in the highway, considering the position of v^* in relation to $q_{i,x}$ and $q_{i,y}$, there will be 3 subcases: (we suppose the x-coordinate $X(q_{i,x})$ of $q_{i,x}$ is less than $X(q_{i,y})$)

1. $X(v^*) \le X(q_{i,x})$. Since the highway Π_{w_x} is legal, we have $d_H(v^*, q_{i,x}) < 2H$. Thus $d_H(v^*, p_{i,x}) \le d_H(v^*, q_{i,x}) + d_H(q_{i,x}, p_{i,x}) \le 2H + d_H(q_{i,x}, p_{i,x})$.
2. $X(q_{i,x}) < X(v^*) \le X(q_{i,y})$. In this case, $d_H(v^*, p_{i,x}) \le \max\{d_H(q_{i,x}, p_{i,x}), d_H(q_{i,y}, p_{i,x})\}$.
3. $X(q_{i,y}) < X(v^*)$. Similar with the preceding subcase 1), we have $d_H(v^*, p_{i,x}) \le 2H + d_H(q_{i,y}, p_{i,x})$.

In summary, we have

$$
\begin{aligned}
d_H(v^*, p_{i,x}) &\le \max\{2H + d_H(q_{i,x}, p_{i,x}), \\
&\quad d_H(q_{i,x}, p_{i,x}), d_H(q_{i,y}, p_{i,x}), \\
&\quad 2H + d_H(q_{i,y}, p_{i,x})\} \\
&= 2H + \max\{d_H(q_{i,x}, p_{i,x}), d_H(q_{i,y}, p_{i,x})\}.
\end{aligned}
$$

Note $d_H(p_{i,x}, q_{i,x}) \le \sqrt{2}c$, and $d_H(p_{i,x}, q_{i,y}) \le d_H(p_{i,x}, p_{i,y}) + d_H(p_{i,y}, q_{i,y}) \le \frac{a}{2^g} + \sqrt{2}c$. Thus, for a sufficiently large n,

$$
d_H(v^*, p_{i,x}) \le 2H + \frac{a}{2^g} + \sqrt{2}c \le (2+\varepsilon)H + \frac{a}{2^g}.
$$

Combining the horizontal span and vertical span of $p_{i,x}$, we know $p_{i,x}$ is in a $h \times \left((2+\varepsilon)H + \frac{a}{2^g}\right)$ rectangle (see Figure 7, case 1).

Case 2: v^* is covered by a horizontal path $E(q_{i,x}, q_{i,y})$ got by Line 16 of Algorithm 3. In this case, the path $E(q_{i,x}, q_{i,y})$ will contain $q_{i,x}, \pi_{z_x,u_1}, \phi_{w_x,u_2}, \phi_{w_x,u_3}, \phi_{z_y,u_4}, q_{i,y}$ in that order.

Considering the position of v^* in this path, there are 3 sub-cases.

Case 2(1) : v^* is covered by a horizontal path $E_1(q_{i,x}, q_{i,y})$ got by line 13 of Algorithm 3. Similar with case 1, $p_{i,x}$ is bounded in the z_x-th horizontal strip. Furthermore.

$$
\begin{aligned}
d_H(v^*, p_{i,x}) &\le 2H + \max\{d_H(q_{i,x}, p_{i,x}), d_H(\pi_{z_x,u_1}, p_{i,x})\} \\
&\le 2H + \max\{\sqrt{2}c, \\
&\quad d_H(p_{i,x}, \phi_{w_x,u_2}) + d_H(\pi_{z_x,u_1}, \phi_{w_x,u_2})\} \\
&\le 2H + H + \sqrt{2}c \le (3+\varepsilon)H
\end{aligned}
$$

So, $p_{i,x}$ is in a rectangle region with height h and width $(3+\varepsilon)H$ (see Figure 7, case 2(1)).

Case 2(2) : v^* is covered by a vertical path $E_2(q_{i,x}, q_{i,y})$ got by line 14 of Algorithm 3. In this case, v^* is on highway Φ_{w_x}, between ϕ_{w_x,u_2} and ϕ_{w_x,u_3}. Since $p_{i,x}$ is on the w_x-th vertical strip, its vertical span is at most h. In addition, $d_V(p_{i,x}, \phi_{w_x,u_3}) \le d_V(p_{i,x}, p_{i,y}) + d_V(p_{i,y}, \pi_{z_y,u_4}) + d_V(\pi_{z_y,u_4}, \phi_{w_x,u_3}) \le \frac{a}{2^g} + H + \sqrt{2}c$, and furthermore, $d_V(p_{i,x}, \phi_{w_x,u_2}) \le d_V(p_{i,x}, \pi_{z_x,u_1}) +$

$d_V(\pi_{z_x,u_1}, \phi_{w_x,u_2}) \le H + \sqrt{2}c$. Similar with case 1, we have

$$
\begin{aligned}
d_V(p_{i,x}, v^*) \\
&\le 2H + \max\{d_V(p_{i,x}, \phi_{w_x,u_2}), d_V(p_{i,x}, \phi_{w_x,u_3})\} \\
&\le 2H + \frac{a}{2^g} + H + \sqrt{2}c \\
&\le (3+\varepsilon)H + \frac{a}{2^g} \text{ (when } n \text{ is large enough)}
\end{aligned}
$$

So, $p_{i,x}$ is in a rectangle region of height h and width $\left((3+\varepsilon)H + \frac{a}{2^g}\right)$ (see Figure 7, case 2(2)).

Case 2(3) : v^* is covered by a horizontal path $E_3(q_{i,x}, q_{i,y})$ got by line 15 of Algorithm 3. In this case, v^* is located in the highway Π_{z_y}. So, $p_{i,y}$ is bounded in the z_y-th horizontal strip. Additionally, we have $d_H(p_{i,y}, \pi_{z_y,u_4}) \le d_H(\pi_{z_y,u_4}, \phi_{w_x,u_3}) + d_H(\phi_{w_x,u_3}, p_{i,x}) + d_H(p_{i,x}, p_{i,y}) \le \sqrt{2}c + H + \frac{a}{2^g}$. Also similar with case 1, we have

$$
\begin{aligned}
d_H(p_{i,y}, v^*) \\
&\le 2H + \max\{d_H(p_{i,y}, \pi_{z_y,u_4}), d_H(p_{i,y}, q_{i,y})\} \\
&\le 2H + \sqrt{2}c + H + \frac{a}{2^g} \\
&\le (3+\varepsilon)H + \frac{a}{2^g} \text{ (when } n \text{ is large enough)}
\end{aligned}
$$

Thus, $p_{i,y}$ is bounded in a rectangle of width $\left((3+\varepsilon)H + \frac{a}{2^g}\right)$ and of height h (see Figure 7, case 2(3)).

In all cases, either $p_{i,x}$ or $p_{i,y}$ is bounded in a rectangle. For some g, the probability that $v*$ is covered by an edge from d is at most $P_g \le h\left((2+\varepsilon)H + \frac{a}{2^g}\right)\frac{4^{g+1}}{a^2} + h(3+\varepsilon)H\frac{4^{g+1}}{a^2} + h\left((3+\varepsilon)H + \frac{a}{2^g}\right)\frac{4^{g+1}}{a^2} + h\left((3+\varepsilon)H + \frac{a}{2^g}\right)\frac{4^{g+1}}{a^2}$, which is $\le h\left(12H + 3\frac{a}{2^g}\right)\frac{4^{g+1}}{a^2}$. Then, consider all $g = 0, 1, 2, \cdots, t-1$, the probability that v^* is covered is at most

$$
\begin{aligned}
p &\le \sum_{g=0}^{t-1} P_g \le \sum_{g=0}^{t-1} h\left(12H + 3\frac{a}{2^g}\right)\frac{4^{g+1}}{a^2} \\
&= 48Hh\sum_{g=0}^{t-1}\frac{4^g}{a^2} + 12h\sum_{g=0}^{t-1}\frac{2^g}{a} \le 48Hh\frac{4^t}{a^2} + 12h\frac{2^t}{a} \\
&\le 48Hh\frac{4k}{a^2} + 12h\frac{2\sqrt{k}}{a} = 192Hh\frac{k}{a^2} + 24h\frac{\sqrt{k}}{a}
\end{aligned}
$$

Replacing H with $\kappa\log\frac{\sqrt{n}}{c\sqrt{2}} - \epsilon_m$ and a with $\sqrt{n}$, we will get

$$
p \le 192\left(\kappa\log\frac{\sqrt{n}}{c\sqrt{2}} - \epsilon_m\right)h\frac{k}{n} + 24h\frac{\sqrt{k}}{\sqrt{n}}
$$

Use the condition $k \leq \theta_3 \frac{n}{\log^2 n}$, we have, for a sufficient large n, $p \leq 192 \left(\kappa \log \frac{\sqrt{n}}{c\sqrt{2}} - \epsilon_m \right) h \frac{\sqrt{k}}{n} \sqrt{\theta_3 \frac{n}{\log^2 n}} + 24 h \frac{\sqrt{k}}{\sqrt{n}} \leq (96 + \varepsilon_1) \kappa h \sqrt{\theta_3} \frac{\sqrt{k}}{\sqrt{n}} + 24 h \frac{\sqrt{k}}{\sqrt{n}} \leq (97 \kappa \sqrt{\theta_3} + 24) h \frac{\sqrt{k}}{\sqrt{n}}$, where ε_1 is a constant that satisfies $0 < \varepsilon_1 \leq 1$. Setting $c_5 = (97 \kappa \sqrt{\theta_3} + 24) h$ finishes the proof. $\square$

With Lemma 14, the following lemma is straightforward.

LEMMA 15. *For any station v^*, the expected number of multicast sessions that pass v^* is at most $c_5 \frac{n_s \sqrt{k}}{\sqrt{n}}$, when $k \leq \theta_3 \frac{n}{\log^2 n}$.*

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

The preceding result only shows the probability upper bound that a given node v^* is used by multicast sessions, when v^* is given a prior. Next, we use VC theorem (Theorem 25) to give the upper bound of the multicast sessions that pass v^* for *every* possible node in the highway system. Recall that, we used n_s sets of k points to generate n_s multicast trees. So, 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 of $\mathcal{C}_{\mathcal{U}}$. Here, we simply use $\log_2 n$ as the upper bound of VC-d($\mathcal{C}_{\mathcal{U}}$). This upper bound is obvious due to the fact that there is 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 $\Theta(\log k)$.

THEOREM 16. *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 14, 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, we have

$$\Pr \left(\sup_{v^*} \left| \frac{\text{\# of sessions covering } v^*}{n_s} - c_5 \frac{\sqrt{k}}{\sqrt{n}} \right| < \epsilon(n) \right) > 1 - \sigma(n)$$

$$\text{if } n_s \geq \max \left\{ \frac{8d}{\epsilon(n)} \cdot \log \frac{13}{\epsilon(n)}, \frac{4}{\epsilon(n)} \log \frac{2}{\sigma(n)} \right\}$$

If we set $\epsilon(n) = \frac{\sqrt{k}}{\sqrt{n}}$ and $\sigma(n) = \frac{2}{n}$, we have

$$\Pr \left(\sup_{v^*}(\text{\# of sessions covering } v^*) < (c_5 + 1) \frac{n_s \sqrt{k}}{\sqrt{n}} \right) > 1 - \frac{2}{n}$$

$$\text{if } n_s \geq \max \left\{ \frac{8\sqrt{n} \log n}{\sqrt{k}} \cdot \log \frac{13\sqrt{n}}{\sqrt{k}}, \frac{4\sqrt{n}}{\sqrt{k}} \log n \right\}$$

$$= \frac{8\sqrt{n} \log n}{\sqrt{k}} \cdot \log \frac{13\sqrt{n}}{\sqrt{k}}$$

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. $\square$

LEMMA 17. *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 1 station in a square of size $c \times c$, we only need to divide the 2nd 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 $B \log \left(1 + \frac{P \cdot \ell(2\sqrt{2}c)}{N_0 + c_3 P(2\sqrt{2}c)^{-\alpha}} \right) / 100$, which is at least a constant.

In addition, w.h.p, each station in highway system is covered by at most $c_6 \frac{n_s \sqrt{k}}{\sqrt{n}}$ multicast sessions when $k \leq \theta_3 \frac{n}{\log^2 n}$. So, the stations of each multicast session can get transmitting rate at least

$$R_2 \geq B \log \left(1 + \frac{P \cdot \ell(2\sqrt{2}c)}{N_0 + c_3 P(2\sqrt{2}c)^{-\alpha}} \right) / \left(100 c_6 \frac{n_s \sqrt{k}}{\sqrt{n}} \right)$$

$$= \frac{B}{100 c_6} \log \left(1 + \frac{P \cdot \ell(2\sqrt{2}c)}{N_0 + c_3 P(2\sqrt{2}c)^{-\alpha}} \right) \frac{\sqrt{n}}{n_s \sqrt{k}}$$

So, if letting $c_7 = \frac{B}{100 c_6} \log \left(1 + \frac{P \cdot \ell(2\sqrt{2}c)}{N_0 + c_3 P(2\sqrt{2}c)^{-\alpha}} \right)$, we get the result we need. $\square$

4.3 Per-flow multicast capacity of the system

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

THEOREM 18. *If $k \leq \theta_1 \frac{n}{\log^{2\alpha+6} n}$ and $n_s \geq \theta_2 n^{1/2+\beta}$, 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 $k \leq \theta_1 \frac{n}{\log^{2\alpha+6} n}$, it is sufficient that $k \leq \theta_3 \frac{n}{\log^2 n}$ for large n. Then both Lemma 11 and Lemma 17 are applicable. We assign the two phases the same amount of time and thus the achievable per-flow date rate is $\frac{1}{2} \min\{c_4, c_7\} \frac{\sqrt{n}}{n_s \sqrt{k}} = c_8 \frac{\sqrt{n}}{n_s \sqrt{k}}$. $\square$

5. UPPERBOUND ON UNICAST CAPACITY

It has been shown in [3] that the asymptotic per-flow unicast capacity under Gaussian channel model is $\Omega(1/\sqrt{n})$ when a square B_n contains a number of nodes following Poisson distribution with rate 1. We here show that the per-flow unicast capacity is at most $O(1/\sqrt{n})$ in this setting.

The basic idea of the proof is as follows. We partition the region into grids of cells of side-length c (value of c depending on p and c_0) such that, the probability that a cell contains at most c_0 nodes is at least p. Here $p > 5/6$ and $c_0 > 1$ are constants. We say a cell is *quasi-closed* if it contains at most c_0 nodes. We call a path of cells *quasi-closed cut* if it contains only quasi-closed cells and crosses from left to right side of square B_n. Furthermore, we define the length of a quasi-closed cut as the total number of cells it contains.

As shown in [3], for all $k > 0$ and $\frac{5}{6} < p < 1$ with $2 + k \log(6(1-p)) < 0$, there exists a number of disjoint groups containing at least $\lceil \delta \log m \rceil$ disjoint paths in every group, and each group is constraint in a stripe of size $m \times (k \log m - \epsilon_m)$, for δ small enough and $\delta \log \frac{p}{1-p} + 1 + k \log(6(1-p)) < 1$, $m = \frac{\sqrt{n}}{c\sqrt{2}}$, and a non-zero small ϵ_m such that the side length of each stripe is integer. When k is some appropriate constant, the number of groups is $\frac{m}{k \log m - \epsilon_m} > n^{1/3}$ when n is large enough. Then by pigeonhole principle, it is easy to show the following

LEMMA 19. *Assume that $\lceil \delta \log m \rceil$ disjoint paths (formed by quasi-closed cells) inside a stripe of length m-cells and width ($k \log m -$*

ϵ_m)-*cells. There exists a quasi-closed cut whose length is at most* $c_9\sqrt{n}$ *for some constant* $c_9 = \frac{k}{\delta c\sqrt{2}}$.

We call such cut in the middle group as $\mathbb{C}$. It is easy to show that the expected number of unicast flows that will cross this cut $\mathbb{C}$ (with end nodes on different sides of $\mathbb{C}$) is at least $n_s/9$, and thus, with high probability, the number of unicast flows that will cross $\mathbb{C}$ is at least $n_s/18$. This implies that, by pigeonhole principle, there exists a cell in $\mathbb{C}$ that will be crossed by at least $\frac{n_s}{18c_9\sqrt{n}}$ unicast flows. Since the cell has only a constant number of nodes, the longest link needed to cross such special cell is at least a constant value ϱ. Thus, the data rate that can be supported by this special link is at most a constant. Thus, we have (whose detailed proofs are omitted due to space limit)

THEOREM 20. *The asymptotic per-flow unicast capacity of* n_s *flows in a large scale random network with* n *nodes randomly distributed in a square* B_n, *with high probability, is at most* $O(\sqrt{n}/n_s)$ *when* n_s *is large enough.*

6. LITERATURE REVIEWS

The ground-breaking work by Gupta and Kumar [7] studied the asymptotic *unicast* capacity of a multi-hop wireless networks for two different models. When each wireless node is capable of transmitting at W bits per second using a constant transmission range, the throughput obtainable by *each* node for a randomly chosen destination is $\Theta(\frac{W}{\sqrt{n\log n}})$ bits per second under PrIM. If nodes are optimally placed and transmission range is optimally chosen, even under optimal circumstances, the throughput is only $\Theta(\frac{W}{\sqrt{n}})$ bits per second for each node. Similar results also hold for PhIM Kulkarni and Viswanath [12] obtained a stronger (almost sure) version of the $\sqrt{n\log n}$ throughput for random node locations in a fixed area obtained in [7].

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. Notice that this is in sharp contrast to the fixed network scenario (when nodes are static after random deployment). In summary, 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 [13]. Gaspar 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.* [21] studied the scaling laws of cognitive networks. Liu *et al.* [15] studied the capacity of a wireless ad hoc network with infrastructure. Another stream of work (*e.g.* [17]) has proposed progressively refined multi-user cooperative schemes, which have been shown to significantly outperform multi-hop 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 multi-channel wireless networks with random (c, f) assignment. Garetto *et al.* [4] studied the capacity scaling in delay tolerant networks with heterogeneous mobile devices. Their methodology allows to identify the scaling laws for a general class of mobile wireless networks, and to precisely determine under which conditions the mobility of nodes can indeed be exploited to increase the per-node throughput.

Broadcast capacity of an arbitrary network has been studied in [9, 20]. They essentially show that, under fPrIM, the broadcast capacity is $\Theta(W)$ for single source broadcast and the achievable broadcast capacity per flow in any network is only $\Theta(W/n)$ if each of the n nodes will serve as source node. This capacity bounds also apply to random networks. Keshavarz-Haddad *et al.* [10] studied the broadcast capacity with dynamic power adjustment for physical interference model. Zheng [23] 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.

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 [14, 19]. Li *et al.* [14] 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/second 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 $k = O(\frac{n}{\log n})$; the per-flow multicast capacity is $\Theta(W/n)$ when $k = \Omega(\frac{n}{\log n})$. Shakkottai *et al.* [19] 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.* [16] studied the multicast capacity for hybrid networks. They derived several capacity regimes based on the relations of the number k of receivers per multicast session, the total number n of nodes, and the number m of base stations.

These results [6–10, 14, 19, 20] for the network capacity of random networks all assumed that the data rate supported by each communication link is a constant W-bps (using PrIM, fPrIM, or PhIM interference models). Using percolation theorem, multihop transmission, pairwise coding and decoding at each hop, and a TDMA scheme, Franceschetti *et al.* [3] shows that a rate $1/\sqrt{n}$ is achievable in networks of randomly located nodes (not only some arbitrarily placed nodes) when Gaussian channel is used.

7. CONCLUSION

In this paper, we studied the multicast capacity of randomly placed wireless nodes in B_n under Gaussian model, in which nodes can transmit data over large distance and the rates of the transmission are determined by SINR. Nodes transmit at constant power P, and the power attenuates according to the power decay law with exponent $\alpha > 2$. We assume that these nodes are randomly located in Poisson distribution of rate 1 in a square B_n with side-length $\sqrt{n}$; there are n_s multicast flows, each flow has k receivers, and the sources and targets of the n_s sessions are chosen by repeating n_s times the process (Algorithm 1). We show that, when $k \leq \theta_1 \frac{n}{(\log n)^{2\alpha+6}}$ and $n_s \geq \theta_2 n^{1/2+\beta}$ for some constants θ_1, θ_2 and any positive real number β, with high probability, each multicast source node can send data to all its intended receivers with rate at least $c_8 \frac{\sqrt{n}}{n_s\sqrt{k}}$ where c_8 is a constant depending on attenuation α, bandwidth B, and background noise N_0. We also present a matching upperbound $O(1/\sqrt{n})$ for per-flow unicast capacity under Gaussian channel.

A number of interesting questions remain open. The first question is to derive an upper bound on the per-flow multicast capacity using the Gaussian channel model for arbitrary k and n. We conjecture that it is also $O(\frac{\sqrt{n}}{n_s\sqrt{k}})$. The second 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 results presented here only hold when $k = O(\frac{n}{(\log n)^{2\alpha+6}})$. We conjecture that

CONJECTURE 21. *When k is $\Omega(\frac{n}{(\log n)^d})$, under Gaussian channel model, the per-flow multicast capacity is $\Theta(\frac{1}{n_s(\log n)^{\alpha/2}})$, for some positive constant $d \le 2\alpha + 6$.*

Notice that, the above conjecture was proved to be true for broadcast (*i.e.* $k = n$) [24] under Gaussian channel model. Their proof is to essentially study the longest edge length of the Euclidean minimum spanning tree, which is asymptotically $\sqrt{\log n}$ when n nodes are randomly placed in a square B_n from [18]. A sufficient condition for the correctness of our conjecture is the correctness of the following conjecture:

CONJECTURE 22. *The longest edge of a Steiner spanning tree, which spans k nodes randomly chosen from n nodes randomly placed in B_n, is of order $\sqrt{\log n}$, when k is large enough, say $\Omega(\frac{n}{\log n})$.*

The third question is to study the capacity when the receiving terminals in a multicast group are within 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.

8. REFERENCES

[1] BHANDARI, V., AND VAIDYA, N. H. Capacity of multi-channel wireless networks with random (c, f) assignment. In *MobiHoc '07* (2007), pp. 229–238.

[2] CHUAH, C.-N., TSE, D. N. C., KAHN, J. M., AND VALENZUELA, R. A. Capacity scaling in MIMO wireless systems under correlated fading. In *IEEE Transactions ON Information Theory* (March 2002), vol. 48.

[3] FRANCESCHETTI, M., DOUSSE, O., N.C.TSE, D., AND THIRAN, P. Closing the gap in the capacity of wireless networks via percolation theory. In *IEEE Transactions ON Information Theory* (2007),

[4] GARETTO, M., GIACCONE, P., AND LEONARDI, E. Capacity scaling in delay tolerant networks with heterogeneous mobile nodes. In *ACM MobiHoc* (2007), pp. 41–50.

[5] GASTPAR, M., AND VETTERLI, M. On the capacity of wireless networks: the relay case. In *IEEE INFOCOM* (2002).

[6] GROSSGLAUSER, M., AND TSE, D. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking (ToN) 10*, 4 (2002), 477–486.

[7] GUPTA, P., AND KUMAR, P. Capacity of wireless networks. *IEEE Transactions on Information Theory IT-46* (1999), 388–404.

[8] JACQUET, P., AND RODOLAKIS, G. Multicast scaling properties in massively dense ad hoc networks. In *ICPADS '05: Proceedings of the 11th IEEE International Conference on Parallel and Distributed Systems* (2005), pp. 93–99.

[9] KESHAVARZ-HADDAD, A., RIBEIRO, V., AND RIEDI, R. Broadcast capacity in multihop wireless networks. In *ACM MobiCom '06* (2006), pp. 239–250.

[10] KESHAVARZ-HADDAD, A., AND RIEDI, R. On the broadcast capacity of multihop wireless networks: Interplay of power, density and interference. In *4th IEEE SECON* (2007).

[11] KESHAVARZ-HADDAD, A., AND RIEDI, R. H. Bounds for the capacity of wireless multihop networks imposed by topology and demand. In *ACM MobiHoc '07* (2007), pp. 256–265.

[12] KULKARNI, S., AND VISWANATH, P. A deterministic approach to throughput scaling in wireless networks. *IEEE Transactions on Information Theory, 50*, 6 (2004), 1041–1049.

[13] LI, J., BLAKE, C., COUTO, D. S. J. D., LEE, H. I., AND MORRIS, R. Capacity of ad hoc wireless networks. In *ACM MobiCom* (2001).

[14] LI, X.-Y., TANG, S.-J., AND OPHIR, F. Multicast capacity for large scale wireless ad hoc networks. In *ACM Mobicom* (2007).

[15] LIU, B., AND TOWSLEY, D. Capacity of a wireless ad hoc network with infrastructure. *ACM MobiHoc* (2007), 239–246.

[16] MAO, X., LI, X.-Y., AND TANG, S.-J. Multicast capacity for hybrid wireless networks. In *ACM MobiHoc* (2008).

[17] ÖZGÜR, A., LÉVÊQUE, O., AND TSE, D. Hierarchical Cooperation Achieves Optimal Capacity Scaling in Ad Hoc Networks. *Information Theory, IEEE Transactions on 53*, 10 (2007), 3549–3572.

[18] PENROSE, M. The random minimal spanning tree in high dimensions. *Annals of Probability 24* (1996), 1903–1925.

[19] SHAKKOTTAI, S., LIU, X., AND SRIKANT, R. The multicast capacity of ad hoc networks. In *Proc. ACM Mobihoc* (2007).

[20] TAVLI, B. Broadcast capacity of wireless networks. *IEEE Communication Letters 10*, 2 (February 2006).

[21] VU, M., DEVROYE, N., SHARIF, M., AND TAROKH, V. Scaling Laws of Cognitive Networks. *Submitted to IEEE Journal on Selected Topics in Signal Processing, June* (2007).

[22] XIE, L., AND KUMAR, P. A network information theory for wireless communication: scaling laws and optimal operation. *Information Theory, IEEE Transactions on 50*, 5 (2004), 748–767.

[23] ZHENG, R. Information dissemination in power-constrained wireless networks. In *IEEE INFOCOM* (2006).

[24] ZHENG, R. Information Dissemination in Power-Constrained Wireless Networks. *IEEE Transaction on Wireless Communication 6*, 12 (2007).

9. APPENDIX

9.1 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 23. *([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*

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

9.2 Legalize the highways

Theorem 6: If we find a set of M horizontal highways and M vertical highways using the above method, we can find a set of M legal horizontal highways and M legal vertical highways.

PROOF. We prove it by legalizing these $2M$ paths.

In the first step, we adjust the horizontal highways, so that any two horizontal highways do not cross each other. As we can see, only the horizontal highways from the same rectangle may cross. If in a rectangle, two highways P_i and P_j across in segments $\overline{\pi_{i,k}\pi_{i,k+1}}$ and $\overline{\pi_{j,l}\pi_{i,l+1}}$, we can break the two highway segments and then build two new highway segments $\overline{\pi_{i,k}\pi_{j,l+1}}$ and $\overline{\pi_{j,l}\pi_{i,k+1}}$. We could see, there are still L highways in the rectangle, but the total length of these L highways is reduced. We repeatedly check whether there are crossing highway segments and break the segments if there are. Since we can not endlessly reduce the total length, at the end of the procedure there are no crossing segments.

Likewise, we can do the same thing with the vertical highways. At the end of the first step, we get a set of M node-disjoint and mutual non-crossing horizontal highways as well as M node-disjoint and mutual non-crossing vertical highways.

In the second step, we try to legalize all the highways. An intuition about a legal highway is that it will never go backward too far, say $2H$ along the coordinate direction.

It seems impossible to legalize a horizontal highway if only a set of M horizontal highways is considered. However, if we take the M vertical highways into account, the goal is achievable. Be aware of that, the M vertical highways are well distributed : each rectangle with width H has L vertical highways.

We try to explain why we can legalize a horizontal highway. For a illegal horizontal highway, it must traverse a vertical rectangle R'_i backwardly. As there are L vertical highways in R'_i, we could find a "shortcut" to reduce the backward interval (notice that a horizontal highway and a vertical highway can share stations). We can do so if we do not result in two horizontal highways sharing a same station. Luckily, we have L vertical highways and we can always legalize the highway while keep all horizontal highways node-disjoint.

To present our method, we consider a illegal horizontal highway Π_i. As what we have shown, it must go backward at least $2H$ and thus horizontally traverse at least 1 vertical rectangle R. Let b be the intersecting point of the backward interval of Π_i and L_l, where L_l is the left side of R. Let a be the last intersecting point before b of Π_i and L_l(see Figure 8). We consider all the cross points of horizontal highways and L_l between a and b, inclusive. We mark these points from up to bottom(we suppose b is under a) with $p_0, p_1, \cdots, p_{t-1}$. It is easy to check that $p_0 = a, p_{t-1} = b$ and t is even. These t points are comprised of $t/2$ pairs of points, with $t/2$ highway intervals connect each pair. Since no two horizontal highways cross each other, there are two kinds of relationship between two highway intervals. Either a highway interval is "inside" another highway interval, or two highway intervals totally exclude each other. A highway interval is 1-level if no highway intervals are inside it. A highway interval is 2-level if it is not 1-level and no highway intervals other than 1-level highway intervals are inside it. Similarly, we call a highway interval is k-level, if it is not 1-level, 2-level, $\cdots$, $k-1$-level, and no highway intervals other than 1-level, 2-level, $\cdots$, $k-1$-level highway intervals are inside it.
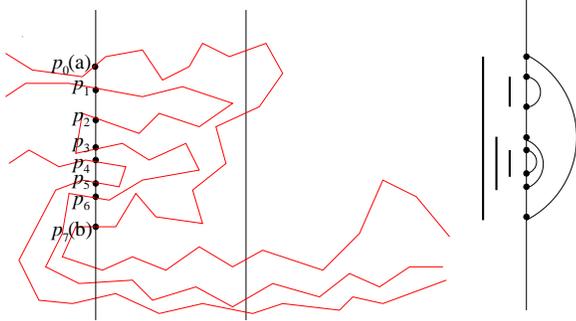
Figure 8 is a example where $t = 8$.

Figure 8: A example of highway intervals when $t = 8$. There are 4 highway intervals, among which there are 2 1-level intervals, 1 2-level interval and 1 3-level interval.

To notice that L vertical highways inside R do not cross each other, we can sort them from left to right. Suppose the L highways are $\Phi_{jL+1}, \Phi_{jL+2}, \Phi_{jL+3}, \cdots, \Phi{jL+L}$ from left to right. We could rebuild these $t/2$ highway intervals without changing the endpoint of them such that for any $0 < k \leq t/2$, and for any $k < l \leq L$, no k-level highway intervals cross with Φ_{jL+l}.

We can do this simply by rebuilding the intervals from low level to high level. Suppose we are now rebuilding the k-level highway

intervals, and all the highway intervals with low levels are rebuilt correctly. The highway Φ_{jL+k} intersects with the k-level highway intervals and divide them into two parts : the part on the left side of the Φ_{jL+k} and the part on the right side of Φ_{jL_k}. Here, "left" and "right" are from the topological view. We keep the left side of these highway intervals, while replace the right side with the highway intervals on Φ_{jL+k}. Then the resulted horizontal highway intervals does not cross with any vertical highway Φ_{jL+l} for $l > k$.(See Figure 9).

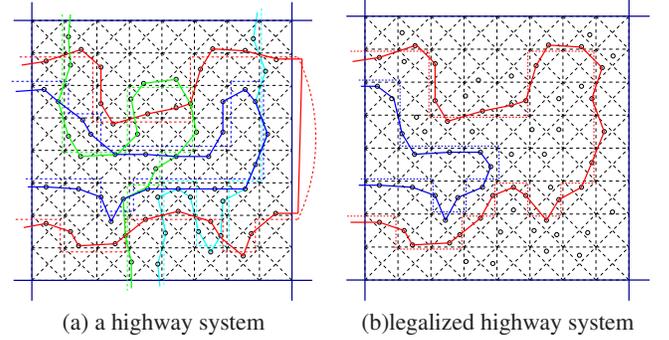
(a) a highway system   (b)legalized highway system

Figure 9: (a) the red line and the blue line are horizontal highway intervals, while the green line and the cyan line are intervals of two vertical highways. (b) the resulted horizontal highway intervals after we legalize the red highway interval.

At the end, we could see that the resulted intervals are all on the left side of $\Phi_{jL+t/2}$, which is or is at left side of Φ_{jL+L}. Therefore, the resulted intervals will not cross with the right side of R.

By repeatedly checking the legality of horizontal highways(vertical highways similarly), finally, we could build a highway network system where all highways are legal. $\square$

9.3 Chernoff bound and VC-theorem

LEMMA 24. *Let X be a Poisson random variable of rate λ.*

$$\Pr(X \geq x) \leq \frac{e^{-\lambda}(e\lambda)^x}{x^x}, \text{for } x > \lambda \qquad (1)$$

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 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 ini the weak law of large numbers:

THEOREM 25 (THE VAPNIK-CHERVONENKIS THEOREM). *If $\mathcal{C}$ is a set of* finite *VC-dimension* VC-d($\mathcal{C}$)*, and $\{X_i \mid i = 1, 2 \cdots, N\}$ is a sequence of i.i.d. random variables with* common *probability distribution P, then for every $\epsilon, \delta > 0$,*

$$\boldsymbol{Pr}\left(\sup_{A \in \mathcal{C}} \left| \frac{\sum_{i=1}^{N} I(X_i \in A)}{N} - \Pr(A) \right| \leq \epsilon \right) > 1 - \delta \qquad (2)$$

whenever $N > \max\left\{ \frac{8 \cdot \text{VC-d}(\mathcal{C})}{\epsilon} \cdot \log \frac{13}{\epsilon}, \frac{4}{\epsilon} \log \frac{2}{\delta} \right\}.$ (3)

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