Multicast Performance With Hierarchical Cooperation

Xinbing Wang, *Member, IEEE*, Luoyi Fu, and Chenhui Hu

***Abstract*—It has been shown in a previous version of this paper that hierarchical cooperation achieves a linear throughput scaling for unicast traffic, which is due to the advantage of long-range concurrent transmissions and the technique of distributed multiple-input–multiple-output (MIMO). In this paper, we investigate the scaling law for multicast traffic with hierarchical cooperation, where each of the n nodes communicates with k randomly chosen destination nodes. Specifically, we propose a new class of scheduling policies for multicast traffic. By utilizing the hierarchical cooperative MIMO transmission, our new policies can obtain an aggregate throughput of $\Omega\left(\left(\frac{n}{k}\right)^{1-\epsilon}\right)$ for any $\epsilon > 0$. This achieves a gain of nearly $\sqrt{\frac{n}{k}}$ compared to the noncooperative scheme in Li *et al.*'s work (*Proc. ACM MobiCom*, 2007, pp. 266–277). Among all four cooperative strategies proposed in our paper, one is superior in terms of the three performance metrics: throughput, delay, and energy consumption. Two factors contribute to the optimal performance: multihop MIMO transmission and converge-based scheduling. Compared to the single-hop MIMO transmission strategy, the multihop strategy achieves a throughput gain of $\left(\frac{n}{k}\right)^{\frac{h-1}{h(2h-1)}}$ and meanwhile reduces the energy consumption by $k^{\frac{\alpha-2}{2}}$ times approximately, where $h > 1$ is the number of the hierarchical layers, and $\alpha > 2$ is the path-loss exponent. Moreover, to schedule the traffic with the converge multicast instead of the pure multicast strategy, we can dramatically reduce the delay by a factor of about $\left(\frac{n}{k}\right)^{\frac{h}{2}}$. Our optimal cooperative strategy achieves an approximate delay-throughput tradeoff $D(n,k)/T(n,k) = \Theta(k)$ when $h \to \infty$. This tradeoff ratio is identical to that of noncooperative scheme, while the throughput is greatly improved.**

***Index Terms*—Capacity, multiple-iput–multiple-output (MIMO), scaling law.**

I. Introduction

CAPACITY of wireless ad hoc networks is constrained by interference between concurrent transmissions. Observing this, Gupta and Kumar adopt the Protocol and the Physical Model to define a successful transmission and study the asymptotically achievable throughput of the network in their seminal work [3]. They show that the per-node throughput capacity scales as $\Theta\left(\frac{1}{\sqrt{n \log n}}\right)$ for random networks, and $\Theta\left(\frac{1}{\sqrt{n}}\right)$ for arbitrary networks, assuming there are n nodes in a unit disk area.

The results on network capacity provide us not only theoretical capacity bounds, but also insights into the protocol design and architecture of wireless networks. Therefore, great efforts are devoted to understanding the scaling laws in wireless ad hoc networks. One important stream of work is improving the unicast capacity. With the percolation theory, Franceschetti *et al.* [4] show that a rate of $\Theta\left(\frac{1}{\sqrt{n}}\right)$ is attainable in random ad hoc networks under the Generalized Physical Model. However, it still vanishes as the number of nodes goes to infinity. To achieve linear capacity scaling, Grossglauser *et al.* [5] exploit nodes' mobility to increase the network throughput, but at the cost of induced delay. Tradeoff between the capacity and the delay is studied in [10]–[12]. An alternative methodology is adding infrastructure to the network. It is shown in [13]–[17] that when the number of base stations grows linearly as that of the nodes (implying a huge investment), the network capacity will scale linearly. Moreover, besides letting nodes perform traditional operations such as storage, replication, and forwarding, [18] and [19] introduce coding into the network, which also brings about the throughput gain.

Another line of research deals with more generalized traffic patterns. Reference [33] studies the scalability of wireless sensor networks. In [20], Toumpis develops asymptotic capacity bounds for nonuniform traffic networks. In [21], broadcast capacity is discussed. Then, a unified perspective on the capacity of networks subject to a general form of information dissemination is proposed in [22]. As a more efficient way for one-to-many data distribution than multiple unicast, multicast fits the applications well such as group communications and multimedia services. Thus, it draws great interest in the research community and has been studied by different manners in [23]–[30]. Specifically, in [24], the authors derive the asymptotic upper and lower bounds for multicast capacity by focusing on data copies and area argument in the routing tree established in the paper. In [25] and [34], multicast capacity is studied under a more realistic channel model, physical-layer model instead of a simplified protocol model assumed in many previous works. In [26], through mathematical derivations and simulations, the authors demonstrate that multicast achieves a gain compared to unicast when information is disseminated to n destinations in mobile ad hoc networks. In [27], a comb-based architecture is proposed instead of a routing tree for multicast, and this is shown to achieve an order-optimal multicast capacity in static networks. In [28], Wang *et al.* prove that network coding cannot necessarily bring about gain in multicast capacity, which is a counterintuitive result. Recently, Niesen *et al.* [31] characterize the multicast capacity region in an extended network. Additionally, capacity-delay tradeoff for mobile multicast is inquired in [32].

Manuscript received February 21, 2010; revised September 15, 2010; April 03, 2011; September 02, 2011; and September 09, 2011; accepted September 17, 2011. This work was supported by the National Basic Research Program of China (973 Program) under Grant 2011CB302701, NSF China under Grant 60832005, the China Ministry of Education Fok Ying Tung Fund under Grant 122002, the China Ministry of Education New Century Excellent Talent under Grant NCET-10-0580, and a Qualcomm Research Grant. An earlier version of this paper appeared in the Proceedings of the IEEE Conference on Computer Communications (INFOCOM), San Diego, CA, March 15–19, 2010.

The authors are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xwang8@sjtu.edu.cn; yiluofu@sjtu.edu.cn; hch@sjtu.edu.cn).

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.2011.2170584

Recently, Aeron *et al.* [6] introduce a multiple-input–multiple-output (MIMO) collaborative strategy achieving a throughput of $\Omega(n^{-1/3})$. Different from Gupta and Kumar's results, they use a cooperative scheme to obtain capacity gain by turning mutually interfering signals into positive factors. Özgür *et al.* [1], [2] utilize hierarchical schemes relying on distributed MIMO communications to achieve linear capacity scaling. The optimal number of hierarchical stages is studied in [7], while multihop and arbitrary networks are investigated in [8] and [9], respectively.

The capacity shown in all the work above is largely bottlenecked by adjacent interference caused by the concurrently transmitting nodes nearby, which is the bottleneck for the capacity of traditional ad hoc networks. This motivates us to investigate multicast scaling laws with hierarchical MIMO in this paper. We jointly consider the effect of traffic patterns and cooperative strategies on the asymptotic performance of networks, aiming to break the bottleneck. To this end, the following fundamental issues should be addressed.

- How to hierarchically schedule multicast traffic to optimize the achievable multicast throughput?
- Is it possible to achieve optimal throughput while maintaining good delay performance and energy efficiency in the network?
- What is the delay-throughput tradeoff with hierarchical cooperative multicast strategies?

To help answer the questions above, we propose a class of hierarchical cooperative scheduling strategies for the multicast traffic. Specifically, we divide the network into clusters; nodes in the same cluster cooperate to transmit data for each other. In this way, all transmissions in the network consist of two parts: intercluster communication and intracluster communication.

A. Intercluster Communication

The transmissions between clusters are conducted by distributed MIMO. When a cluster acts as a sender, all nodes in the cluster transmit a *distinct* bit at the same time. Then, each node in the receiving cluster can observe a signal containing information of all transmitted bits.

We propose two kinds of transmission method: direct and multihop MIMO transmission, which are more general than that in [35]. For the communication between clusters, the direct method uses MIMO transmission only once from the source cluster to all destination clusters, while the multihop method conducts MIMO transmissions for many hops, with each time a cluster only transmits to the neighboring cluster. We will show in this paper that multihop MIMO transmission can increase the throughput and reduce the energy consumption due to better spatial reuse and power management.

B. Intracluster Communication

To decode MIMO transmissions, the destination nodes in each destination cluster must observe results from all other nodes in the same cluster. Since each cluster may act as a destination cluster of multiple source clusters, there are several sets of destination nodes in it. For each set, every node in the cluster sends one *identical* bit to all nodes in the set. This traffic can be seen as multicast, but considering the "converge" nature of the data flows, it can also be regarded as *converge multicast*. Hence, we propose two kinds of scheduling strategies: multicast-based strategy and converge-based strategy.

Comparing these two kinds of strategies, we find that they make no difference in terms of throughput and energy consumption of the network. However, the converge-based strategy can dramatically reduce the delay by approximately $\Theta\big((\frac{n}{k})^{\frac{h}{2}}\big)$, where $h > 1$ is the number of hierarchical layers in the network. We further divide the cluster into "subclusters" and still use distributed MIMO to realize the communication between each other. When using multicast-based strategy, each source node must distribute data within its subcluster first, which accounts for the major part of the delay. In contrast, utilizing the converge nature of the traffic, converge-based strategy omits the distribution procedure and significantly reduces the delay.

Our main contributions are as follows.

- We propose a class of hierarchical cooperative scheduling policies for the multicast traffic, which can achieve the throughput close to the information-theoretic upper bound.
- We reschedule the traffic of our cooperative transmission and dramatically reduce the delay.
- The cooperative multicast scheme proposed in this paper greatly improves the network throughput, while it achieves the same delay-throughput tradeoff as the noncooperative multicast scheme, and the cooperative multicast tradeoff even outperforms that of unicast in some special cases.

Main results in the paper are as follows.[1]

- We achieve a throughput of $\widetilde{\Theta}\big((\frac{n}{k})^{\frac{2h-2}{2h-1}}\big)$, which has a gain of nearly $\sqrt{\frac{n}{k}}$ compared to the noncooperative scheme.
- The delay of our optimal strategy is $\widetilde{\Theta}\big(n^{\frac{2h-4}{2h-1}}k^{\frac{3}{2h-1}}\big)$, which achieves a delay-throughput tradeoff ratio $\widetilde{\Theta}\big(k(\frac{k}{n})^{\frac{2}{2h-1}}\big)$.
- The energy-per-bit consumption is $O\big(n^{\frac{1-\alpha}{2h-1}}k^{-\frac{2h\alpha-3\alpha+2}{4h-2}}\big)$.

The remainder of this paper is organized as follows. In Section II, we introduce our network models and definitions of terms. In Section III, we outline the multicast hierarchical cooperative scheme. The analysis of throughput, delay, and energy consumption are presented in Sections IV, V-A, and V-B, respectively. All the results are discussed in detail in Section VI. Finally, we conclude the paper in Section VII.

II. Network Models and Definitions

A. Network Models

We consider a set of n nodes in $V = \{v_1, v_2, \ldots, v_n\}$ uniformly and independently distributed in a unit square Ω. Each node v_i acts as a source node of a multicast session.

Multicast Traffic: For a source node v_i, we randomly and independently choose a set of k nodes in $U_i = \{u_{i,j} | 1 \le j \le k\}$ other than v_i in the deployment square as its destination nodes. We define a multicast *session* as the collection of transmissions from one source node to k destination nodes and use $\mathrm{MP}(n, k)$ to denote an n-session multicast problem with each node acting as a source node for a session.

We then define another traffic that helps our analysis.

Converge Multicast Traffic: We randomly and independently choose a set of k nodes $U_i = \{u_{i,j} | 1 \le j \le k\}$ as destinations.

[1]We use Knuth's notation in this paper. Also, we use $f(n) = \widetilde{\Theta}(g(n))$ to indicate $f(n) = O(n^{\epsilon} g(n))$ and $f(n) = \Omega(n^{-\epsilon} g(n))$, for any $\epsilon > 0$. Intuitively, this means $f(n) = \Theta(g(n))$ with logarithmic terms ignored.

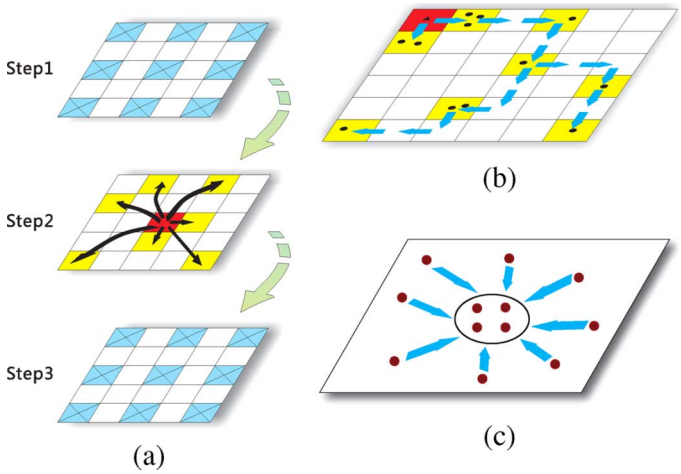


Fig. 1. Transmission strategy of hierarchical cooperation. (a) Three-step structure. (b) Multihop MIMO transmission. (c) Converge multicast transmission frame.

Each of the n nodes in the network acts as a source node and sends one identical bit to all nodes in U_i. This is a "converge" transmission because the overall data flow is from all n nodes to the set of k nodes, as shown in Fig. 1(c); we define it as a converge multicast *frame*. Denoting $\mathrm{CMP}(n, m, k)$ as an m-frame converge multicast problem, for each frame we choose a set of k destination nodes.

Wireless Channel Model: We assume that the communication is over a channel of limited bandwidth W. Each node has a power budget of P. For the transmission from v_j to v_i, the channel gain between them at time t is given by

$$g_{ij}[t] = \sqrt{G} d_{ij}^{-\alpha/2} e^{j\theta_{ij}[t]} \tag{1}$$

where d_{ij} is the distance between v_i and v_j, $\theta_{ij}[t]$ is the random phase at time t, uniformly distributed in $[0, 2\pi)$. $\{\theta_{ij}[t] | 1 \le i, j \le n\}$ is a collection of independently and identically distributed (i.i.d.) random processes. The parameters G and $\alpha > 2$ are constants, where α is called the path-loss exponent. The signal received by node v_i at time t thus can be expressed as

$$Y_i[t] = \sum_{j \in \mathbb{T}[t]} g_{ij}[t] X_j[t] + Z_i[t] + I_i[t] \tag{2}$$

where $Y_i[t]$ is the signal received by node v_i at time t, $\mathbb{T}[t]$ represents the set of active senders that can be added constructively, $Z_i[t]$ is the Gaussian noise at node v_i of variance N_0 per symbol, and $I_i[t]$ is the interference from the nodes which are destructive to the reception of node v_i.

When conducting the cooperative transmission, we assume that the full channel state information (CSI) is available at each node.[2] Also, we assume the far-field condition holds for all nodes, i.e., the minimum distance between any two nodes is larger than the wavelength of the carrier.[3]

In this paper, we only consider the *dense network*, which means that the network area is a unit square. Our hierarchical cooperative scheme can also be applied to the *extended network*, with a $\sqrt{n} \times \sqrt{n}$ square network area.

[2]This assumption is also made in [1].

[3]The assumption is proved to be reasonable in the first paragraph of [1, p. 3].

B. Definition of Performance Metrics

Definition of Throughput: A per-node throughput of $\lambda(n, k)$ bits/s is feasible if there is a spatial and temporal transmission scheme, such that every node can send $\lambda(n, k)$ bits/s on average to its k randomly chosen destination nodes. The aggregate multicast throughput of the system is $T(n, k) = n\lambda(n, k)$. When $k = 1$, it degenerates to aggregate unicast throughput.

Definition of Delay: The delay $D(n, k)$ of a communication scheme for the network is defined as the average time it takes for a bit to reach its k destination nodes after leaving its source node. The averaging is over all bits transmitted in the network.

Definition of Energy-Per-Bit: We introduce energy-per-bit $E(n, k)$ to define the average energy required to carry one bit from a source node to one of its k destination nodes.

III. Transmission Strategy

A. General Multicast Structure

The key idea of our multicast structure is dividing the network into *clusters* with equal numbers of nodes, then the traffic can be transformed into intra- and intercluster transmissions. In this way, we divide the network into two *layers*: the clusters and the whole network. We call the former *lower layer*, and the latter *upper layer*, and let n_1 and n_2 be the number of nodes in the lower and upper layer, respectively. In each multicast session, there is a source node as well as k randomly chosen destination nodes. Let k_1 be the number of destination nodes in a cluster, and $k_2 = k$ be that in the network. We term the cluster containing the source node *source cluster*, and clusters containing at least one destination node *destination clusters*. Each multicast session is realized by a three-step structure [see Fig. 1(a)].

Step 1) The source node distributes n_1 bits to n_1 nodes in the cluster, one bit for each node. The traffics in this step are unicasts from the source node to $n_1 - 1$ other nodes in the same cluster.

Step 2) The nodes in the source cluster transmit simultaneously by implementing *distributed MIMO transmission* to convey data to the destination clusters. There are two means for MIMO transmissions.

- *Multihop MIMO transmission:* Each source cluster uses MIMO to transmit to a neighboring cluster called *relay cluster*. After each node in the relay cluster receives a MIMO observation, it amplifies the received signal to a desirable power and retransmits it to the following relay cluster in the next chance according to the routing protocol. This process is repeated until all the destination clusters receive MIMO observations, as shown in Fig. 1(b).
- *Direct MIMO transmission:* The nodes in the source cluster *broadcast* the data in the network simultaneously, and then all nodes in the destination clusters can receive a MIMO observation.

Step 3) After each destination cluster receives the MIMO transmissions, each node in the cluster holds an observation. The k_1 destination nodes in the cluster must collect all n_1 observations to decode the n_1 transmitted bits. Thus, the traffics in this step are n_1 multicast sessions, with each node in the cluster acting as a source node. Also, the k_1 destination

nodes are identical for all n_1 sessions. Hence, the traffic can also be treated as a *converge multicast problem*, where all source nodes "converge"[4] their data to a set of destination nodes.

Following the steps above, we can build a hierarchical scheme in a network with multiple layers to achieve the desired throughput. At the lowest layer, we use simple TDMA protocol to exchange bits for setting up cooperation among small clusters. Combining this with multihop MIMO transmissions, we get a higher throughput scheme for cooperation among nodes in larger clusters at the next layer. Finally, at the top layer, the size of the cooperation clusters is maximum, and the MIMO transmissions are almost over the global scale to meet the desired traffic demands.

B. Four Strategies for Cooperative Multicast

Following the three-step multicast structure, we propose four strategies that can realize the steps. A *multilayer solution* is involved in each of the strategies.

- Multihop MIMO multicast (MMM): Step 3 is formulated as a multicast problem with multihop MIMO transmissions. The multicast delivery in Step 3 can also be handled using the same three-step structure. Implementing the three-step structure recursively, we can get a hierarchical solution to the multicast problem.
- Direct MIMO multicast (DMM): Step 3 is formulated as a multicast problem with direct MIMO transmissions.
- Converge-based multihop MIMO multicast (CMMM): Step 3 is formulated as a converge multicast problem with multihop MIMO transmissions, and the converge multicast problem can also be solved with the multilayer manner.
- Converge-based direct MIMO multicast (CDMM): Step 3 is formulated as a converge multicast problem with direct MIMO transmissions.

For the hierarchical schemes with multiple layers, we give the more detailed definition of the converge multicast frame introduced in the CMMM and CDMM schemes as following.

1) Converge Multicast: Consider the cooperative hierarchical scheme with two layers. At layer i, for any destination cluster, there are n_{i-1} nodes in that cluster, with k_{i-1} of them being destinations. The converge multicast frame here refers to the traffic pattern where all the n_{i-1} nodes in this destination cluster transmit their data to those k_{i-1} destinations. Here, there are n_1 multicast sessions, with each node in the cluster acting as a source node.

C. Notations

We use the following notations throughout this paper. First, let h be the number of layers that is independent of n and k. Then, we label each layer with a unique number i ($1 \le i \le h$), indicating the ith layer from the bottom up.

Given a layer i, let n_i be the number of nodes and k_i be that of destination nodes for each source node; apparently, $n_h = n$ and $k_h = k$. We use $n_{c_i} = n_i/n_{i-1}$ to denote the number of clusters, and k_{c_i} the number of destination clusters at layer i.

When analyzing strategies, we use m_i to denote the number of multicast sessions at layer i when considering the MMM/DMM, or the number of converge multicast frames at layer i when considering the CMMM/CDMM.

[4]Note that the traffic mode is similar to converge-cast in Step 3. Our multicast analysis can well cover converge-cast case, where sources transmit information to the destination with distinctive data rates.

IV. Analysis of Multicast Throughput

In this section, we first present the information-theoretic upper bound of the multicast throughput. Then, we provide strategies that can almost achieve the upper bound by utilizing cooperation in the network. When analyzing the throughput, we use an "assume-and-verify" method, i.e., we first make some assumptions on the network; after we obtain the results, we verify these assumptions. Using this method, we make our analysis both strict and easy to follow.

A. Upper Bound of Multicast Throughput

To prove the upper bound, we need to set the lower bound of the pairwise distance between nodes, which is provided in the following lemma.

Lemma 4.1: In a network with n nodes randomly and uniformly distributed on a unit-square, the minimum distance between any two nodes is $\frac{1}{n^{1+\delta}}$ *w.h.p.*,[5] for any $\delta > 0$.

Proof: Consider a specific node v_i, proving the distance between v_i and all other nodes is larger than $\frac{1}{n^{1+\delta}}$ is equivalent to proving that there are no other nodes inside a circle of area $\frac{\pi}{n^{2+2\delta}}$ around v_i. The probability of such an event is $\left(1-\frac{\pi}{n^{2+2\delta}}\right)^{n-1}$. The minimum distance between any two nodes in the network is larger than $\frac{1}{n^{1+\delta}}$ only if this condition is satisfied for all nodes in the network. Thus, by union bound we have

$$P\left[d_{ij} \le \frac{1}{n^{1+\delta}},\ \text{for all } i,j \text{ and } i \ne j\right] \le n\left(1-\left(1-\frac{\pi}{n^{2+2\delta}}\right)^{n-1}\right)$$

which diminishes to zero when n tends to infinity. ■

Theorem 4.1: In the network with n nodes and each sending packets to k randomly chosen destination nodes, the aggregate multicast throughput is bounded by

$$T(n,k) \le p_1 \frac{n \log n}{k}$$

w.h.p., where $p_1 > 0$ is a constant independent of n and k.

Proof: For each source node in the network, we have randomly assigned k destination nodes to it. If the sets of destination nodes for each source node do not intersect with each other, there will be totally nk nodes acting as destination nodes. However, there are only n nodes in the whole network. Thus, by considering the source–destination pairing from a reverse view, for each node d, there are k nodes $s_1, s_2, \ldots, s_k$ on average, which choose d as one of its destination nodes. Assume each source node transmits data to d at the same rate $\lambda(n,k)$, the total rate $k\lambda(n,k)$ from the source nodes s_i ($1 \le i \le k$) to the destination node d is upper-bounded by the capacity of a multiple-input–single-output (MISO) channel between d and the rest of the network. Using a standard formula for this channel, we get

$$k\lambda(n,k) \le \log\left(1+\frac{P}{N_0}\sum_{\substack{i=1\\ s_i\ne d}}^{n}|g_{s_i d}|^2\right) = \log\left(1+\frac{P}{N_0}\sum_{\substack{i=1\\ s_i\ne d}}^{n}\frac{G}{d_{s_i d}^{\alpha}}\right).$$

[5]In this paper, "*w.h.p.*" stands for "with high probability," which means the probability tends to 1 as $n \to \infty$.

According to Lemma 4.1, the distance between s_i ($1 \le i \le k$) and d is larger than $\frac{1}{n^{1+\delta}}$ *w.h.p.* With this fact, we obtain

$$\lambda(n,k) \le \frac{1}{k}\log\left(1+\frac{GP}{N_0}n^{\alpha(1+\delta)+1}\right) \le \frac{p_1 \log n}{k}$$

w.h.p. for some constant p_1 independent of n and k. The theorem then follows. ∎

B. Throughput Analysis With MMM

To ensure successful MIMO transmissions, each cluster must have the same number of nodes. The following lemma ensures the number of nodes in each cluster at layer $2 \le i \le h$ has the same order. For simplicity, we consider the number of nodes in each cluster is exactly n_{i-1}.

Lemma 4.2: Consider n_i nodes uniformly distributed in the network area. We divide the network into n_{c_i} identical square-shaped clusters. Then, the number of nodes in each cluster is $n_{i-1} = \frac{n_i}{n_{c_i}}$ *w.h.p.* when *Assumption 1:* $n_i = \Omega(n_{c_i}\log n_{c_i})$ is satisfied.

Proof: The number of nodes in a cluster at layer i is the sum of i.i.d. Bernoulli random variables X_j, such that $P[X_j = 1] = 1/n_{c_i}$. Using Chernoff bounds

$$P\left[\sum_{j=1}^{n_i} X_j \ge (1+\delta)\frac{n_i}{n_{c_i}}\right] < e^{-f(\delta)\frac{n_i}{n_{c_i}}}$$

with $f(\delta) = (1+\delta)\log(1+\delta) - \delta$

$$P\left[\sum_{j=1}^{n_i} X_j \le (1-\delta)\frac{n_i}{n_{c_i}}\right] < e^{-\frac{1}{2}\delta^2\frac{n_i}{n_{c_i}}}.$$

When $n_i = \Omega(n_{c_i}\log n_{c_i})$

$$P\left[\left|\sum_{j=1}^{n_i} X_j - \frac{n_i}{n_{c_i}}\right| \ge \delta\frac{n_i}{n_{c_i}}\right] < e^{-\frac{n_i}{n_{c_i}}\theta} \to 0$$

if $n \to +\infty$. Here, $\theta > 0$ is a constant depending only on δ, thus $n_{i-1} = \sum_{j=1}^{n_i} X_j = \Theta(\frac{n_i}{n_{c_i}})$ *w.h.p.* ∎

Remark 4.1: Note that the purpose of Lemma 4.2 is to show the relationship between the number of nodes at layer i, denoted by n_i, and the number of cells at layer i, namely, n_{c_i}. Actually, how n_{c_i} varies at each layer depends on not only n, but also the number of total layers h and the property of the cooperative scheme adopted as well. Different hierarchical divisions at each layer will lead to different throughput results. In our following MMM, CMMM, DMM, and CDMM schemes, the detailed dependency of n_{c_i} on n can be revealed during the analysis on throughput and delay.

As mentioned earlier, we solve the $\mathrm{MP}(n,k)$ in the network area with a three-step approach. Since the problems in Steps 1 and 3 are also multicast problems,[6] we can apply the three steps recursively and build an h-layer solution.

1) Solution to Multicast Problem: We consider the ith layer in the network ($2 \le i \le h$) and follow the three steps.

Step 1: Preparing for Cooperation: Given the total number of multicast sessions m_i at layer i, each node holds $\frac{m_i}{n_i}$ bits to multicast. In this step, each node must distribute all its data to other nodes in the same cluster, with $\frac{m_i}{n_i n_{i-1}}$ bits for each. As there are n_{i-1} source nodes in each cluster, the traffic load is $\Theta\left(\frac{m_i n_{i-1}}{n_i}\right)$ bits. Since the data exchanges only involve intracluster communication, they can work according to the 9-TDMA scheme. We divide time into slots; at each time slot, let the neighboring eight clusters keep silent when the centric cluster is exchanging data. According to the channel model (2), we assume that the received interference signal $I_r(t)$ is a collection of uncorrelated zero-mean stationary and ergodic random processes with power upper-bounded by a constant.[7] This assumption is also adopted in the proof of Lemma 3.1 [2]. Thus, the power of destructive interference is bounded, enabling clusters to operate simultaneously according to the 9-TDMA scheme, which is ensured by Lemma 4.3.

Lemma 4.3: By the 9-TDMA scheme, when $\alpha > 2$, one node in each cluster has a chance to exchange data at a constant transmission rate. Also, when $\alpha > 2$, the interfering power received by a node from the simultaneously active clusters is upper-bounded by a constant.

Proof: We divide the network into groups, each of which contains nine subsquares. The nine squares in each group are numbered from 1 to 9 the same way as in the 9-TDMA scheme. We further divide time into sequences of successive slots, denoted by t ($t = 0, 1, 2, 3, \ldots$). During a particular slot t, one node in subsquares that are numbered $(t \bmod 9) + 1$ is allowed to transmit packets.

In a slot, if a node inside the subsquare s_i is allowed to transmit to another node inside s_i, those nodes that may interfere with the current transmission must be located along the perimeters of concentric subsquares centered at s_i. The interfering nodes can be grouped based on their distances to s_i such that the jth group contains $8j$ interfering nodes or less (near the boundary of the network) and the shortest distance from the receiver in s_i is $(3j-1)\sqrt{A}$, where A is the area of the subsquare. Assuming that all nodes use the same transmission power $P(n,k)$, with the power propagation model in (1), the cumulative interference at subsquare s_i, denoted by I_{s_i}, can be bounded by

$$\begin{aligned} I_{s_i} &\le \sum_{j=1}^{n/M} 8j \times \frac{GP(n,k)}{[(3j-1)\sqrt{A}]^\alpha} \\ &\le \frac{8GP(n,k)}{A^{\frac{\alpha}{2}}}\left[1+\sum_{j=2}^{n/M}(3j-1)^{1-\alpha}\right] \\ &< \frac{8GP(n,k)}{A^{\frac{\alpha}{2}}}\left[1+\int_{j=0}^{\infty}(3j+2)^{(1-\alpha)}dj\right] \\ &< \frac{8GP(n,k)}{A^{\frac{\alpha}{2}}}\left[1+\frac{1}{3(\alpha-2)}\right] \\ &= \frac{8GP(n,k)}{A^{\frac{\alpha}{2}}}\cdot\frac{3\alpha-5}{3\alpha-6}. \end{aligned} \tag{3}$$

If we choose the transmission power $P(n,k) = \Theta(A^{\frac{\alpha}{2}})$, the interfering power will be upper-bounded by a constant independent of n. Since the maximum distance for a transmitter

[6] We view unicast as a special case of multicast problem.

[7] This assumption is also needed in other strategies, so we will not repeat. Also note that negligible channel interference is one of the basic catches that make both our work and analysis go through. Without the guarantee of constant bounded interference, we cannot ensure the high decoding probability at the receiving nodes.

to a receiver is $\sqrt{2A}$, the reception power can be lower-bounded by

$$R_{s_i} \geq \frac{GP(n,k)}{(\sqrt{2A})^{\alpha}}. \tag{4}$$

As a result, the signal-to-interference-plus-noise ratio (SINR) for the transmission in s_i, denoted by SINR_{s_i}, is

$$\begin{aligned}\mathrm{SINR}_{s_i} &= \frac{R_{s_i}}{N_0 + I_{s_i}} \\ &\geq \frac{\frac{GP(n,k)}{(\sqrt{2A})^{\alpha}}}{N_0 + \frac{8GP(n,k)}{A^{\frac{\alpha}{2}}} \cdot \frac{3\alpha-5}{3\alpha-6}}.\end{aligned} \tag{5}$$

Note that $P(n,k) = \Theta(A^{\frac{\alpha}{2}})$, and the SINR is a constant irrelative to n and k. Therefore, a fixed transmission rate independent of n and k can be achieved, according to Shannon's channel capacity formula, i.e., $R(n,k) = W\log(1+\mathrm{SINR})$, where $R(n,k)$ is the feasible rate and W is the channel bandwidth. ■

Under the assumption of an aggregate unicast throughput of $\Theta(n_{i-1}^a), 0 \leq a \leq 1$ can be achieved for every possible source–destination pair at layer $(i-1)$. Given a traffic load of $\Theta\big(\frac{m_i n_{i-1}}{n_i}\big)$ bits, this step can be completed in $\Theta\Big(\frac{m_i n_{i-1}^{1-a}}{n_i}\Big)$ time slots.

Step 2: Multihop MIMO Transmissions: In this step, each source cluster starts a series of MIMO transmissions to reach all its corresponding destination clusters using the multihop method. To achieve the asymptotically optimal multicast throughput, we construct a multicast tree (MT) by adopting [26, Algorithm 1], spanning over a source cluster S_i and its corresponding destination clusters D_{ij}, where $1 \leq j \leq k_{c_i}$. Algorithm 1 is briefly described as follows to make this paper self-contained.

For a set of nodes $P_i = \{S_i, D_{ij}, 1 \leq j \leq k_{c_i}\}$ containing a super source node and its super destination nodes, we first build a Euclidean spanning tree, denoted as $\mathrm{EST}(P_i)$, to connect them. For each link uv in $\mathrm{EST}(P_i)$, we decompose it into a Manhattan path connecting u and v to form Manhattan routing tree $\mathrm{MRT}(P_i)$. Then, for each edge uw in $\mathrm{MRT}(P_i)$, we connect super nodes crossed by uw in sequence. The final tree is called multicast tree MT.

The constructed MT owns the following properties, with which we can acquire by Lemma 4.4.

- *The maximum length of each hop at layer i is* $\Theta\big(\sqrt{\frac{n_{i-1}}{n}}\big)$.
- *The total length of* $\mathrm{MT}(P_i)$ *is at most* $O\big(\sqrt{k_{c_i}} \times \sqrt{\frac{n_i}{n}}\big)$.

Lemma 4.4: The number of hops in the MT is $O\Big(\sqrt{\frac{n_i k_{c_i}}{n_{i-1}}}\Big)$.

For all m_i multicast sessions, at layer i there are $\frac{m_i}{n_{i-1}}$ MTs, and the total number of hops is $O\Big(\frac{m_i}{n_{i-1}}\sqrt{\frac{n_i k_{c_i}}{n_{i-1}}}\Big)$. Using the 9-TDMA scheduling, each cluster is allowed to take MIMO transmission in every nine time slots. If a cluster serves as a relay cluster for multiple multicast sessions, it will deliver the packets of different sessions including its own packets with equal probability. According to our protocol, clusters can transmit simultaneously at each time slot $\Theta\big(\frac{n_{c_i}}{9}\big)$. Therefore, the total amount of time to accomplish all m_i sessions' MIMO transmissions is $O\Big(m_i\sqrt{\frac{k_{c_i}}{n_i n_{i-1}}}\Big)$.

Step 3: Cooperative Decoding: Now that each MT has k_{c_i} destination clusters, after Step 2, every cluster receives $\Theta\big(\frac{m_i k_{c_i}}{n_i}\big)$ MIMO transmissions.[8] For each MIMO transmission, every node in a destination cluster obtains an observation that the n_{i-1} bits are transmitted from the source node. To decode the original n_{i-1} bits, all nodes in the destination cluster must quantify each observation into Q bits, where Q is a constant. After that, each node conveys the Q bits to all k_{i-1} destination nodes in the cluster. Obviously, this procedure can be formulated as an $\mathrm{MP}(n_{i-1}, k_{i-1})$. After all observation results reach the destination nodes, they can decode the transmitted n_{i-1} bits.

Assume that an aggregate multicast throughput $\widetilde{\Theta}(n_{i-1}^a k_{i-1}^b)$ is achievable at layer $(i-1)$ *w.h.p.*, where $0 \leq a \leq 1$, $-1 \leq b \leq 0$, and $a+b \leq 0$, then the $\mathrm{MP}(n_{i-1}, k_{i-1})$ can be solved within $\widetilde{\Theta}\Big(\frac{Qn_{i-1}}{n_{i-1}^a k_{i-1}^b}\Big)$ time slots. Note that each cluster receives $\Theta\big(\frac{m_i k_{c_i}}{n_i}\big)$ MIMO transmissions and needs to perform this decoding process for each transmission. By utilizing the 9-TDMA scheme, we can finish all $m_{i-1} = m_i k_{c_i}$ multicast sessions in $\Theta\big(\frac{m_i k_{c_i}}{n_i}\big)$ rounds. Consequently, Step 3 costs $\widetilde{\Theta}\Big(\frac{m_i n_{i-1}^{1-a} k_{c_i}}{n_i k_{i-1}^b}\Big)$ time slots.

Finally, the transmission should be performed at the bottom layer. Since every node broadcasts its data in each session, all destination nodes can receive one bit, and a multicast session can be completed in one time slot.

2) Division of the Network: By minimizing the total time cost during the three steps at layer i, we present the throughput-optimal division of the network.

Lemma 4.5: Given k_i independently and uniformly distributed destination nodes in the network at layer i, the number of destination clusters k_{c_i} is given by

$$k_{c_i} = \begin{cases} \Theta(k_i), & \text{when } k_i = O(n_{c_i}) \\ \Theta(\frac{n_i}{n_{i-1}}), & \text{when } k_i = \Omega(n_{c_i}). \end{cases}$$

Proof: Let X_j be a random variable

$$X_j = \begin{cases} 1, & \text{if cluster } j \text{ contains at least one destination node} \\ 0, & \text{else} \end{cases}$$

then $k_{c_i} = \sum_{j=1}^{n_{c_i}} X_j$. Since the k_i destination nodes at layer i are uniformly and independently distributed in n_{c_i} clusters, the probability that a destination node is in cluster j is $1/n_{c_i}$, and the probability that none of the k_i destination nodes is in cluster j is $\big(1-\frac{1}{n_{c_i}}\big)^{k_i}$, which gives

$$E[X_j] = 1 - \Big(1 - \frac{1}{n_{c_i}}\Big)^{k_i}.$$

Since $\{X_j\}_1^{n_{c_i}}$ is a sequence of i.i.d. random variables, using the law of large numbers, we obtain *w.h.p.* that

$$\frac{k_{c_i}}{n_{c_i}} = \frac{1}{n_{c_i}}\sum_{j=1}^{n_{c_i}} X_j \to 1 - \Big(1 - \frac{1}{n_{c_i}}\Big)^{k_i} \text{ when } n_{c_i} \to \infty. \tag{6}$$

Consequently, the number of clusters that contain at least one destination node is $k_{c_i} = n_{c_i}\big(1-(1-\frac{1}{n_{c_i}})^{k_i}\big)$. If $k_i = O(n_{c_i})$,

[8] This is valid under Assumption 3 in Lemma 4.7, which we present later.

$k_{c_i} = n_{c_i}\big(1-(1-\frac{1}{n_{c_i}})^{k_i}\big) = \Theta(k_i)$ *w.h.p.*; if $k_i = \Omega(n_{c_i})$, $k_{c_i} = n_{c_i}\big(1-(1-\frac{1}{n_{c_i}})^{k_i}\big) = \Theta(n_{c_i})$ *w.h.p.* ■

Lemma 4.6: When *Assumption 2:* $m_h = O\big((n_{c_i})^{p_2}\big)$ holds for all $2 \le i \le h$ with a constant $p_2 > 0$:

(a) if $k_i = \Omega(n_{c_i}\log n_{c_i})$, then $k_{i-1} = \Theta\big(\frac{k_i}{n_{c_i}}\big)$ *w.h.p.*;

(b) if $k_i = O(n_{c_i}\log n_{c_i})$, then $k_{i-1} = O(\log n_{c_i})$ *w.h.p.*

In Lemma 4.7, we use l_i to denote the number of destination sets in each cluster. More specifically, let each source node choose a set of destination nodes in the network, and l_i be the number of source nodes that choose at least one destination node in the layer i of the network. We have $l_i = m_i/\prod_{j=i+1}^{h} n_{c_j}$ for MMM/DMM, in which m_i is the number of multicast sessions, and $l_i = m_i/\prod_{j=i+1}^{h-1} n_{c_j}$ for CMMM/CDMM, in which m_i is the number of converge multicast frames.

Lemma 4.7: When $k_i = o(n_{c_i})$, the number of destination sets at the $(i-1)$th layer l_{i-1} is:

(a) when *Assumption 3:* $l_i = \Omega\big(\frac{n_{c_i}}{k_i}\log\frac{n_{c_i}}{k_i}\big)$ is satisfied, then $l_{i-1} = \Theta\big(\frac{l_i k_i}{n_{c_i}}\big)$ *w.h.p.*;

(b) when $l_i = O\big(\frac{n_{c_i}}{k_i}\log\frac{n_{c_i}}{k_i}\big)$, then $l_{i-1} = O\big(\log\frac{n_{c_i}}{k_i}\big)$ *w.h.p.*

Now we are ready to present our network division scheme.

Lemma 4.8: When $k = O(n^{1-\epsilon})$ for a small $\epsilon > 0$, the number of nodes at each layer to achieve optimal throughput with the MMM strategy is

$$n_i = \begin{cases} (\frac{n}{k})^{\frac{2i-1}{2h-1}}, & i < h \\ n, & i = h. \end{cases} \tag{7}$$

Proof: Still, we consider the three steps at layer i. When Assumptions 1 and 3 are satisfied, combining the three steps, the total time to complete m_i multicast sessions is

$$\Theta\left(\frac{m_i n_{i-1}^{1-a}}{n_i}\right) + O\left(m_i\sqrt{\frac{k_{c_i}}{n_i n_{i-1}}}\right) + \widetilde{\Theta}\left(\frac{m_i n_{i-1}^{1-a}k_{c_i}}{n_i k_{i-1}^{b}}\right). \tag{8}$$

Since the time cost of Step 3 is always higher than that of Step 1 in order of magnitude, the throughput at layer i is given by

$$\begin{aligned} T(n_i,k_i) &= \frac{m_i}{\Theta\left(\frac{m_i n_{i-1}^{1-a}}{n_i}\right)+O\left(m_i\sqrt{\frac{k_{c_i}}{n_i n_{i-1}}}\right)+\widetilde{\Theta}\left(\frac{m_i n_{i-1}^{1-a}k_{c_i}}{n_i k_{i-1}^{b}}\right)} \\ &= \widetilde{\Theta}\left(\frac{n_i n_{i-1}}{\sqrt{n_i n_{i-1}k_{c_i}} + n_{i-1}^{2-a}k_{i-1}^{-b}k_{c_i}}\right). \end{aligned} \tag{9}$$

To optimize the network division at layer i, we consider two cases: $n_{c_i} = O(k_i)$ and $n_{c_i} = \Omega(k_i)$.[9] We suppose Assumption 2 is satisfied, then the properties of two cases are summarized as follows, according to Lemmas 4.5 and 4.6.

Case 1) When $n_{c_i} = O(k_i)$, then $k_{c_i} = \Theta(n_{c_i})$, $k_{i-1} = \widetilde{\Theta}\big(\frac{k_i}{n_{c_i}}\big)$.

Case 2) When $n_{c_i} = \Omega(k_i)$, then $k_{c_i} = \Theta(k_i)$, $k_{i-1} = O(\log n_{c_i}) = \widetilde{\Theta}(1)$.

In Case 1, the throughput in (9) can be

$$T(n,k) = \widetilde{\Theta}\left(\frac{n_i n_{i-1}}{n_i + n_{i-1}^{1-a-b}k_i^{-b}n_i^{1+b}}\right) \tag{10}$$

which is optimized when $n_{i-1} = (\frac{n_i}{k_i})^{\frac{b}{1-a-b}}$. However, since Case 1 requires that $n_{c_i} = O(k_i)$, or $n_{i-1} = \Omega(\frac{n_i}{k_i})$, the optimal result cannot be achieved. Thus, the maximum achievable throughput in Case 1 is $\widetilde{\Theta}\Big(\frac{n_i}{k_i+n_i^{1-a}k_i^{a}}\Big)$ when we choose $n_{i-1} = n_i/k_i$, which is not superior to the throughput at the $(i-1)$th layer.

In Case 2, the throughput in (9) can be

$$T(n,k) = \widetilde{\Theta}\left(\frac{n_i n_{i-1}}{\sqrt{n_i k_i/n_{i-1}} + n_{i-1}^{2-a}k_i}\right) \tag{11}$$

which is optimized when $n_{i-1} = (\frac{n_i}{k_i})^{\frac{1}{3-2a}}$. Since the inequality $(\frac{n_i}{k_i})^{\frac{1}{3-2a}} < \frac{n_i}{k_i}$ holds, we can achieve a throughput of $\widetilde{\Theta}\big((\frac{n_i}{k_i})^{\frac{2-a}{3-2a}}\big)$, which is better than the throughput at the $(i-1)$th layer as $0 \le a < 1$. Therefore, we can improve the throughput by adopting Case 2.

At the bottom layer, the aggregate multicast throughput is $T(n_1,k_1) = 1$. When network is divided in the optimal way at each layer, the relationship among n_i, k_i and throughput in each layer is as follows:

$$\begin{aligned} n_h &= k_h n_{h-1}^{\frac{2h-1}{2h-3}}, \quad \Big(\frac{n_h}{k_h}\Big)^{\frac{2h-2}{2h-1}} \\ &\vdots \\ n_3 &= n_2^{\frac{5}{3}}, \quad \Big(\frac{n_3}{k_3}\Big)^{4/5} \\ n_2 &= n_1^{3}, \quad \Big(\frac{n_2}{k_2}\Big)^{2/3}. \end{aligned} \tag{12}$$

Substituting $n_h = n, k_h = k$ into (12), we obtain the results in (7). This finishes the proof. ■

Remark 4.2: Now, the number of sessions at each layer is $m_i = n\prod_{j=i+1}^{h} k_j = \widetilde{\Theta}(nk)$. Under this condition, when (7) is satisfied, the time spent at each layer is in the same order of magnitude, i.e., it takes the same amount of time for the broadcast transmission at the bottom layer and for the multihop MIMO transmission at every other layer. However, when $m_i = \Theta(nk)$ does not exactly hold, the throughput of the network is determined by the layer with the maximum number of sessions $\max_{1\le i\le h-1}\{m_i\}$. This conclusion also holds for the CMMM strategy, with m_i denoting the number of frames at each layer. To get the precise throughput result of the MMM strategy, we must further calculate the number of multicast sessions at each layer.

3) Verification of Assumptions: To calculate the accurate throughput result, there are three conditions that need justification. We now consider these factors under (7).

a) First we consider Assumptions 1 and 2. With the multicast traffic pattern described earlier, the number of multicast sessions at the top layer is $m_h = n$, which is smaller than $n_{c_i}^h$ for $2 \le i \le h$, thus Assumption 2 holds. As for Assumption 1, it is obvious that $k_i = O(\log n_{c_{i+1}}) = O(\frac{n_i}{\log n_{c_i}})$ for $1 \le i \le h-1$, and $k = O(\frac{n}{n_{c_h}})$ if $k = O(n/\log^{\frac{2h-1}{2h-3}} n)$ at the top layer. Since we only consider the case $k = O(n^{1-\epsilon})$ for a small $\epsilon > 0$, Assumption 1 is also satisfied.

b) Then, we consider the number of destination nodes at each layer. By Lemma 4.6

$$k_i = \begin{cases} O\big(\log(\frac{n}{k})^{\frac{2}{2h-1}}\big), & 1 \le i \le h-2 \\ O\big(\log k(\frac{n}{k})^{\frac{2}{2h-1}}\big), & i = h-1. \end{cases}$$

[9]The network division is equivalent to power control. By optimal network division, a node does not need to transmit with full power. This can well solve the problem of limited power.

This will change the number of sessions to

$$\begin{aligned} m_h &= n \\ m_{h-1} &= nk \\ m_{h-2} &= nk \log\left(\frac{n}{k}\right) \\ &\vdots \\ m_1 &= nk \log^{h-2}\left(\frac{n}{k}\right). \end{aligned} \tag{13}$$

c) In our scheme, Lemma 4.7(a) must be applied recursively, and we have to ensure Assumption 3 is satisfied at each recursion. Recall that the number of destination sets is

$$l_i = \frac{m_i}{\prod_{j=i+1}^{h} n_{c_j}} = \frac{m_i}{k(n/k)^{\frac{2h-2i}{2h-1}}}$$

combining (13), we obtain $l_i = \Omega\big((\frac{n}{k})^{\frac{2}{2h-1}}\big)$. Note that in our network division $\frac{n_{c_i}}{\log n_{c_i}} \log \frac{n_{c_i}}{\log n_{c_i}} = \Theta\big((\frac{n}{k})^{\frac{2}{2h-1}}\big)$ for $2 \le i \le h-1$, thus $l_i = \Omega\big(\frac{n_{c_i}}{k_i} \log \frac{n_{c_i}}{k_i}\big)$, and Assumption 3 is satisfied for all layers.

4) Calculation of Throughput: From the analysis above, plus the conclusion of Remark 4.2, the throughput is determined by the number of sessions at the bottom layer because $m_1 = \max_{1\le i\le h-1}\{m_i\} = nk\log^{h-2}(\frac{n}{k})$. Followed by (9), the throughput is

$$T(n,k) = \Theta\bigg(\Big(\frac{n}{k}\Big)^{\frac{2h-2}{2h-1}} \log^{-(h-2)} \frac{n}{k}\bigg). \tag{14}$$

Then, the following theorem naturally holds.

Theorem 4.2: By using the MMM strategy, we can achieve an aggregate throughput of

$$T(n,k) = \Theta\bigg(\Big(\frac{n}{k}\Big)^{\frac{2h-2}{2h-1}} \log^{-(h-2)} \frac{n}{k}\bigg). \tag{15}$$

C. Throughput Analysis With the CMMM

Consider three top layers h, $h-1$, and $h-2$, with layer $h-1$ and $h-2$ termed as "clusters" and "subclusters," respectively. We organize $\frac{n_{h-1}}{n_{h-2}}$ *rounds* of transmission, and choose a subcluster in every cluster for each round ($\frac{n_h n_{h-2}}{n_{h-1}}$ source nodes per round). Only nodes in the chosen subclusters serve as source nodes at each round, and each round is divided into three steps.

Step 1: Preparing for Cooperation: Each source node in the chosen subclusters must deliver n_{h-1} bits to nodes in the same cluster for cooperation, one bit for each node. This includes two substeps.

- *Substep 1: MIMO Transmissions:* In a specific cluster, each node acts as a destination node. For each destination node d, the chosen subcluster uses direct MIMO transmission[10] to communicate with the subcluster where d locates. This takes n_{h-1} time slots to accomplish.
- *Substep 2: Cooperate Decoding:* All subclusters in the network perform decoding in parallel, which makes this substep a $\mathrm{CMP}(n_{h-2}, n_{h-2}, 1)$.

[10]Because the time cost in Step 1 is not the dominating factor on throughput, this will not affect the result. The reason we do not use multihop is that the traffic is not uniformly distributed and is hard to schedule by TDMA scheme.

Step 2: Multihop MIMO Transmission: After Step 1, all source nodes in the chosen subcluster have distributed their n_{h-1} bits to the nodes in the same cluster. To use multihop MIMO transmission, we must build $\frac{n_h n_{h-2}}{n_{h-1}}$ MTs, with each corresponding to a source node. According to Lemma 4.4 and the 9-TDMA scheme, Step 2 can be completed in $\Theta\Big(n_{h-2}\sqrt{\frac{n_h k_{c_h}}{n_{h-1}}}\Big)$ time slots.

Step 3: Cooperative Decoding: Each destination cluster works in parallel and decodes the original n_{h-2} bits from MIMO observations. The decoding process can be formulated as a $\mathrm{CMP}(n_{h-1}, m_{h-1}, k_{h-1})$, with $m_{h-1} = n_{h-2}k_{c_h}$. This conclusion is based on Assumption 3.

1) Solution to Converge Multicast Problem: We start by studying a two-layer network. Given a $\mathrm{CMP}(n_2, m_2, k_2)$, we divide the network into clusters of n_1 nodes, where a frame of transmission includes the following steps.

Step 1: After the division of clusters, there are k_{c_2} destination clusters. Since all n_2 nodes must send one bit to k_2 destination nodes, all n_{c_2} clusters must act as source clusters and transmit to k_{c_2} destination clusters using MIMO.

For each of the n_{c_2} source clusters, build an MT connecting the source and destination clusters. By Lemma 4.4, we can finish all the transmissions on MTs in $O\Big(\sqrt{\frac{n_2 k_{c_2}}{n_1}}\Big)$ slots. Considering m_2 frames, the time cost in Step 1 is $O\Big(m_2\sqrt{\frac{n_2 k_{c_2}}{n_1}}\Big)$.

Step 2: After a destination cluster receives a MIMO transmission, all n_1 nodes must quantify the observation and converge them to the destination nodes in the cluster, which is a converge multicast problem. When Assumption 3 is satisfied, there are $m_1 = \Theta\big(\frac{m_2 k_{c_2}}{n_{c_2}}\big)$ frames that choose a cluster as the destination cluster. Thus, there is a $\mathrm{CMP}(n_1, m_1, k_1)$ to handle in each cluster.

Since the problem in Step 2 is also a converge multicast problem, our two-step scheme can be applied recursively to construct a hierarchical solution. In our CMMM strategy, we build an $(h-1)$-layer strategy for Step 3, plus the top layer, hence there is a total of h layers.

At last, we specify the transmission of the bottom layer. For each frame, every node broadcasts its data, and all destination nodes can receive one bit per time slot. The frame can be completed in n_1 time slots.

2) Division of the Network: Similar to MMM strategy, we first present the throughput-optimal network division.

Lemma 4.9: When $k = O(n^{1-\epsilon})$ for a small $\epsilon > 0$, the number of nodes at each layer to achieve optimal throughput in the CMMM strategy is

$$n_i = \begin{cases} (\frac{n}{k})^{\frac{2i-1}{2h-1}}, & i < h \\ n, & i = h. \end{cases} \tag{16}$$

Proof: We consider two layers i and $i-1$, with $2 \le i \le h-1$. As the $\mathrm{CMP}(n_{i-1}, m_{i-1}, k_{i-1})$ at the $(i-1)$th layer can be solved in $\widetilde{\Theta}(m_{i-1}n_{i-1}^a k_{i-1}^b)$ time slots, and we assume that Assumptions 1–3 are satisfied as in the analysis of the MMM strategy, we then have $m_{i-1} = \Theta(m_i k_{c_i})$. Still, we consider two cases: $n_{c_i} = O(k_i)$ and $n_{c_i} = \Omega(k_i)$, with the properties still holding.

Case 1) When $n_{c_i} = O(k_i)$, then $k_{c_i} = \Theta(n_{c_i})$, $k_{i-1} = \Theta\big(\frac{k_i}{n_{c_i}}\big)$.

Case 2) When $n_{c_i} = \Omega(k_i)$, then $k_{c_i} = \Theta(k_i)$, $k_{i-1} = O(\log n_{c_i}) = \widetilde{\Theta}(1)$.

In case 1, the $\mathrm{CMP}(n_i, m_i, k_i)$ can be solved in

$$m_i\sqrt{\frac{n_i k_{c_i}}{n_{i-1}}} + m_{i-1}n_{i-1}^a k_{i-1}^b = \frac{m_i n_i}{n_{i-1}} + m_i n_i^{1-b}k_i^b n_{i-1}^{a+b-1} \tag{17}$$

time slots. The result is optimized by choosing $n_{i-1} = (\frac{n_i}{k_i})^{\frac{b}{a+b}}$. However, $(\frac{n_i}{k_i})^{\frac{b}{a+b}} < \frac{n_i}{k_i}$, which contradicts with the requirement $n_{i-1} = \Omega(\frac{n_i}{k_i})$ of Case 1. Thus, the minimum time to solve the $\mathrm{CMP}(n_i, m_i, k_i)$ is $m_i n_i^a k_i^{1-a}$, which is achieved when $n_{i-1} = n_i/k_i$. This is not superior to the solving time at the $(i-1)$th layer.

In Case 2, the $\mathrm{CMP}(n_i, m_i, k_i)$ can be solved in

$$m_i\sqrt{\frac{n_i k_{c_i}}{n_{i-1}}} + m_{i-1}n_{i-1}^a k_{i-1}^b = m_i\sqrt{\frac{n_i k_i}{n_{i-1}}} + m_i k_i n_{i-1}^a \tag{18}$$

time slots. The result is optimized by choosing $n_{i-1} = (\frac{n_i}{k_i})^{\frac{1}{2a+1}}$. Since $(\frac{n_i}{k_i})^{\frac{1}{2a+1}} < \frac{n_i}{k_i}$ holds, $\mathrm{CMP}(n_i, m_i, k_i)$ can be solved in $mn_{i+1}^{\frac{a}{2a+1}}k_{i+1}^{\frac{a+1}{2a+1}}$ time slots, which is better than the solving time at the ith layer. Therefore, we can reduce the solving time by adopting Case 2.

At the bottom layer, a frame can be finished in n_1 time slots. When we divide the network in the optimal way at each layer, the relationship of n_i, k_i and the solving time in each layer from 1 to $h-1$ is shown as follows:

$$\begin{aligned} n_{h-1} &= k_{h-1}n_{h-2}^{\frac{2h-3}{2h-5}}, \quad n_{h-1}^{\frac{1}{2h-3}}k_{h-1}^{\frac{2h-4}{2h-3}} \\ &\vdots \\ n_3 &= n_2^{\frac{5}{3}}, \quad n_3^{1/5}k_3^{4/5} \\ n_2 &= n_1^3, \quad n_2^{1/3}k_2^{2/3}. \end{aligned} \tag{19}$$

Thus, the minimum solving time of $\mathrm{CMP}(n_{h-1}, m_{h-1}, k_{h-1})$ is $\widetilde{\Theta}\Big(m_{h-1}n_{h-1}^{\frac{1}{2h-3}}k_{h-1}^{\frac{2h-4}{2h-3}}\Big)$.

With all procedures put together, we deliver $n_{h-1} \times n_{h-2} \times \frac{n_h}{n_{h-1}}$ bits to their destination nodes in

$$\Big(n_{h-1}+n_{h-2}^{\frac{2h-6}{2h-5}}\Big)+n_{h-2}\sqrt{\frac{n_h k_{c_h}}{n_{h-1}}}+n_{h-2}k_h n_{h-1}^{\frac{1}{2h-3}}k_{n-1}^{\frac{2h-4}{2h-3}} \tag{20}$$

time slots at every round of transmission. Therefore, the aggregate throughput is

$$\frac{n_{h-1} \times n_{h-2} \times \frac{n_h}{n_{h-1}}}{\Big(n_{h-1}+n_{h-2}^{\frac{2h-6}{2h-5}}\Big)+n_{h-2}\sqrt{\frac{n_h k_{c_h}}{n_{h-1}}}+n_{h-2}k_h n_{h-1}^{\frac{1}{2h-3}}k_{n-1}^{\frac{2h-4}{2h-3}}} \tag{21}$$

which can be optimized by choosing $n_{h-1} = (\frac{n_h}{k_h})^{\frac{2h-3}{2h-1}}$. Combining with (19), we obtain (16). This finishes the proof. ■

3) Verification of Assumptions: Before presenting the throughput result, the three conditions in Section IV-B.3 also need justification.

a) To begin with, the verification procedure of Assumptions 1 and 2 and the assumptions are the same as that for the MMM strategy, so we omit the details here.

b) Then, we consider the number of destination nodes at each layer. Since $m_{h-1} = k(\frac{n}{k})^{\frac{2h-5}{2h-1}}$ and $k_{i-1} = \log n_{c_i}$ for $2 \le i \le h$

$$\begin{aligned} m_{h-1} &= k\Big(\frac{n}{k}\Big)^{\frac{2h-5}{2h-1}} \\ m_{h-2} &= k\Big(\frac{n}{k}\Big)^{\frac{2h-5}{2h-1}}\log\Big(\frac{n}{k}\Big) \\ &\vdots \\ m_1 &= k\Big(\frac{n}{k}\Big)^{\frac{2h-5}{2h-1}}\log^{h-2}\Big(\frac{n}{k}\Big). \end{aligned} \tag{22}$$

c) In our scheme, Lemma 4.7(a) must be applied recursively. We need to ensure that Assumption 3 is satisfied at each recursion. Recall that the number of destination sets is

$$l_i = \frac{m_i}{\prod_{j=i+1}^{h-1} n_{c_j}} = \frac{m_i}{(n/k)^{\frac{2h-2i-2}{2h-1}}}$$

and the equation still holds under the network division (16). Combining (22), we obtain $l_i = \Omega\big((\frac{n}{k})^{\frac{2}{2h-1}}\big) = \Omega\big(\frac{n_{c_i}}{\log n_{c_i}}\log\frac{n_{c_i}}{\log n_{c_i}}\big)$ for $3 \le i \le h-1$, and Assumption 3 is satisfied. However, when $i = 2$

$$l_2 = k\Big(\frac{n}{k}\Big)^{\frac{1}{2h-1}}\log^{h-3}\Big(\frac{n}{k}\Big).$$

Comparing l_2 to $(\frac{n}{k})^{\frac{2}{2h-1}}$, we find there exists a threshold[11]

$$k_{\mathrm{th}} = \Theta\big(n^{\frac{1}{2h}}\log^{\frac{(h-3)(2h-1)}{2h}} n\big) = \widetilde{\Theta}\big(n^{\frac{1}{2h}}\big). \tag{23}$$

When $k = \Omega(k_{\mathrm{th}})$, Assumption 3 holds for layer 2; otherwise it does not. Thus the number of frames at the bottom layer is

$$m_1 = \begin{cases} (\frac{n}{k})^{\frac{2h-4}{2h-1}}\log(\frac{n}{k}), & \text{when } k = O(k_{\mathrm{th}}), \\ k(\frac{n}{k})^{\frac{2h-5}{2h-1}}\log^{h-2}(\frac{n}{k}), & \text{when } k = \Omega(k_{\mathrm{th}}). \end{cases} \tag{24}$$

4) Calculation of Throughput: With the analysis above and the conclusion of Remark 4.2, the throughput is determined by the number of frames at the bottom layer because $m_1 = \max_{1\le i\le h-1}\{m_i\}$. Thus, followed by (21) and (24), the throughput is given by

$$T(n,k) = \begin{cases} \Theta\big(n^{\frac{2h-3}{2h-1}}k^{\frac{2}{2h-1}}\log^{-1}\frac{n}{k}\big), & \text{when } k = O(k_{\mathrm{th}}) \\ \Theta\big((\frac{n}{k})^{\frac{2h-2}{2h-1}}\log^{-(h-2)}\frac{n}{k}\big), & \text{when } k = \Omega(k_{\mathrm{th}}) \end{cases} \tag{25}$$

based on which we have the following theorem.

Theorem 4.3: With the CMMM strategy, we can achieve an aggregate throughput of

$$T(n,k) = \begin{cases} \Theta\big(n^{\frac{2h-3}{2h-1}}k^{\frac{2}{2h-1}}\log^{-1}\frac{n}{k}\big), & \text{when } k = O(n^{\frac{1}{2h}}) \\ \Theta\big((\frac{n}{k})^{\frac{2h-2}{2h-1}}\log^{-(h-2)}\frac{n}{k}\big), & \text{when } k = \Omega(n^{\frac{1}{2h}}). \end{cases} \tag{26}$$

[11]We will discuss the influence of it in Section VI-C.

D. Broadcast Case

So far, we have only proven the throughput result when $k = O(n^{1-\epsilon})$ for an arbitrarily small $\epsilon > 0$. Another case is $k = \widetilde{\Theta}(n)$, which we refer to as the *broadcast case*.

According to Theorem 4.2, the network cannot be divided into more than $\widetilde{\Theta}(n_i)$ clusters at layer i. Therefore, we can only divide the network as $n_{c_i} = O(k_i)$ for the broadcast case. This division has been discussed in the proof of Lemmas 4.8 and 4.9 (see Case 1), and the throughput performance does not increase as the number of layers becomes larger. Consequently, there is no gain on the throughput when utilizing our cooperative scheme in the broadcast case, and the throughput results in Theorems 4.2 and 4.3 still hold.

In the rest of this paper, we do not distinguish $k = O(n^{1-\epsilon})$ and $k = \widetilde{\Theta}(n)$ because the conclusions hold for both cases.

E. Throughput Analysis With Direct MIMO Transmission

The DMM and CDMM operate in the similar way to the MMM and CMMM, respectively. The only difference is that we use direct MIMO transmission in the former two strategies. Due to the similarity, we only present some important conclusions and results under the DMM and CDMM strategies.

In the DMM and CDMM, we perform direct MIMO transmissions at each layer, which takes one time slot for each source cluster. This difference leads to another optimized network division for both DMM and CDMM

$$n_i = \begin{cases} (\frac{n}{k})^{\frac{i}{h}}, & i < h \\ n, & i = h. \end{cases} \tag{27}$$

Under this division, the throughput results are given by the following theorem.

Theorem 4.4: With either the DMM or the CDMM strategy, we can achieve an aggregate throughput

$$T(n,k) = \Theta\left(\left(\frac{n}{k}\right)^{\frac{h-1}{h}} \log^{-(h-1)} \frac{n}{k}\right). \tag{28}$$

V. Delay and Energy Consumption Analysis

A. Delay Analysis

1) Delay Analysis With the MMM: As mentioned in Section IV, delay performance of the MMM is poor. Intuitively, at the ith layer, a source node must divide the data into n_{i-1} parts of the same size and distribute to other nodes for cooperation. This division is repeated at each layer. Since the smallest part of data at the bottom later is one bit, the minimum size of data packets at layer i is $B_i = \prod_{j=1}^{i-1} n_j$ bits.

For the ith layer, let $D(n_i, k_i)$ be the average time to accomplish a multicast session for each of n_i nodes. To analyze the delay, we consider the three steps separately.

1) For Step 1, each source node distributes B_i bits to other nodes within the same cluster. Because in this step, all traffic is unicast, the distribution process takes $D(n_{i-1}, 1)$ time slots. We ignore the time spent in Step 1 since it is smaller than that in Step 3.
2) For Step 2, to transmit B_i bits for all n_i source nodes, there are $n_i B_i / n_{i-1}$ MTs at layer i. The number of hops on each MT at layer i is $\Theta\big(\sqrt{\frac{n_i k_i}{n_{i-1}}}\big)$. Using the 9-TDMA scheme, we can accomplish $\Theta(\frac{n_i}{n_{i-1}})$ hops per time slot, and thus can complete the second step in $\Theta\big(B_i\sqrt{\frac{n_i k_i}{n_{i-1}}}\big)$ time slots.
3) For Step 3, the traffic loads are $n_{i-1}k_i$ multicast sessions in every cluster. Recall that we use $D(n_{i-1}, k_{i-1})$ to denote the amount of time to finish the transmission of n_{i-1} multicast sessions at layer $i-1$; therefore, Step 3 takes $k_i D(n_{i-1}, k_{i-1})$ time slots.

These three steps cost $D(n_i, k_i)$ time slots, thus

$$D(n_i, k_i) = \Theta\left(B_i\sqrt{\frac{n_i k_i}{n_{i-1}}}\right) + k_i D(n_{i-1}, k_{i-1}) \tag{29}$$

where $B_i = (\frac{n}{k})^{\frac{(i-1)^2}{2i-1}}$ for $1 \le i \le h$. For the bottom-layer transmission scheme, $D(n_1, k_1) = n_1 = (\frac{n}{k})^{\frac{2}{2h-1}}$. Substituting these into (29) and iterating the equation for $i = 1, 2, \ldots, h$, we then obtain the final result

$$D(n,k) = \Theta\Big(n^{\frac{h^2-2h+2}{2h-1}} k^{-\frac{h^2-4h+3}{2h-1}}\Big). \tag{30}$$

Remark 5.1: According to the result above, the delay is determined by the number of nodes at each layer, and the transmission time at the top layer is the dominating factor on delay, which implies that we can just calculate the time cost at the top layer.

Combining (15) with (30), we obtain the delay-throughput tradeoff

$$D(n,k)/T(n,k) = \Theta\Big(n^{\frac{h^2-4h+3}{2h-1}} k^{-\frac{h^2-6h+4}{2h-1}} \log^{h-2}\frac{n}{k}\Big). \tag{31}$$

2) Delay Analysis With the CMMM: In the CMMM strategy, the delay is the amount of time a transmission round spends, and it is calculated in the throughput analysis. The time cost to finish each round is given by (20). By Lemma 4.9, substituting all parameters with n and k in (20), we obtain the delay

$$D(n,k) = \begin{cases} \Theta\big((\frac{n}{k})^{\frac{2h-3}{2h-1}} \log\frac{n}{k}\big), & \text{when } k = O(k_{\text{th}}) \\ \Theta\big(n^{\frac{2h-4}{2h-1}} k^{\frac{3}{2h-1}} \log^{h-2}\frac{n}{k}\big), & \text{when } k = \Omega(k_{\text{th}}) \end{cases} \tag{32}$$

which is simplified as

$$D(n,k) = \begin{cases} \widetilde{\Theta}\big((\frac{n}{k})^{\frac{2h-3}{2h-1}}\big), & \text{when } k = O(n^{\frac{1}{2h}}) \\ \widetilde{\Theta}\big(n^{\frac{2h-4}{2h-1}} k^{\frac{3}{2h-1}}\big), & \text{when } k = \Omega(n^{\frac{1}{2h}}). \end{cases} \tag{33}$$

Combining (33) with (26), we find the delay-throughput tradeoff is

$$\frac{D(n,k)}{T(n,k)} = \begin{cases} \Theta\left(k^{-1}\log\frac{n}{k}\right), & \text{when } k = O(n^{\frac{1}{2h}}) \\ \Theta\big(k(\frac{n}{k})^{-\frac{2}{2h-1}} \log^{h-2}\frac{n}{k}\big), & \text{when } k = \Omega(n^{\frac{1}{2h}}). \end{cases} \tag{34}$$

3) Delay Analysis With the DMM: The delay analyzing procedure of DMM is similar to that of MMM. Thus, we can easily obtain the delay result by the conclusion of Remark 5.1.

For DMM, each time a source node must transmit $B_h = (\frac{n}{k})^{\frac{h-1}{2}}$ bits. The transmission rate at the top layer is $n^{\frac{1}{h}} k^{\frac{h-1}{h}}$ bit/s using MIMO. Then, we derive the delay as

$$D(n,k) = \Theta\Big(n^{\frac{h^2-h+2}{2h}} k^{\frac{h^2-3h+2}{2h}}\Big). \tag{35}$$

Combining with (28), the delay-throughput tradeoff is

$$D(n,k)/T(n,k) = \Theta\Big(n^{\frac{h^2-3h+4}{2h}} k^{-\frac{h^2-5h+4}{2h}} \log^{h-1}\frac{n}{k}\Big). \tag{36}$$

4) Delay Analysis With the CDMM: The way we obtain the delay of the CDMM is similar to that of the CMMM. The result is

$$D(n,k) = \Theta\Big(n^{\frac{h-1}{h}} k^{\frac{1}{h}} \log^{h-1}\frac{n}{k}\Big). \tag{37}$$

Compared to (30), the CMMM strategy reduces the delay dramatically by a factor of nearly $(\frac{n}{k})^{\frac{h}{2}}$. Combining (37) with (28), we obtain the delay-throughput tradeoff

$$D(n,k)/T(n,k) = \Theta\left(k\log^{2h-2}\frac{n}{k}\right). \tag{38}$$

B. Energy Consumption Analysis

Suppose that the energy consumption for each transmission is proportional to d^α, where d is the distance between the sender and the receiver and $\alpha > 2$ is the path-loss exponent. Recall that we define $E(n,k)$ as the energy cost to carry one bit from a source node to one of its k destination nodes. We focus on the energy consumption of the MMM strategy and present only the results for the rest of the three strategies, as all the results can be obtained in the similar way.

1) Energy Consumption of the MMM: In the MMM strategy, a multicast session is divided into three steps, and we study the three steps one after another. For the ith layer, we use $E(n_i,k_i)$ to denote the energy consumption.

1) For Step 1, each source node distributes packets within the network. The amount of traffic load is less than that in Step 3 in order of magnitude, so we need not consider the power spent in this step.
2) For Step 2, the number of hops on each MT are $\Theta\big(\sqrt{\frac{n_i k_i}{n_{i-1}}}\big)$. For each hop, all n_{i-1} nodes in the sending cluster must transmit to a distance of $\sqrt{\frac{n_{i-1}}{n}}$, which is the side length of a cluster at the ith layer. The energy consumed in the transmissions on each MT is
$$O\left(n_{i-1}\sqrt{\frac{n_i k_i}{n_{i-1}}}\Big(\frac{n_{i-1}}{n}\Big)^{\frac{\alpha}{2}}\right).$$
3) For Step 3, we will perform $\Theta(n_{i-1}k_i)$ sessions of multicast at layer $(i-1)$, with each transmitting Qk_{i-1} bits. Hence, the energy consumption in this step is
$$O\big(n_{i-1}k_i E(n_{i-1},k_{i-1})\big).$$

In these three steps, a total of $n_{i-1}k_i$ bits are transmitted. According to the above analysis

$$\begin{aligned} &n_{i-1}k_i E(n_i,k_i) \\ &\quad = n_{i-1}\sqrt{\frac{n_i k_i}{n_{i-1}}}\Big(\frac{n_{i-1}}{n}\Big)^{\frac{\alpha}{2}} + n_{i-1}k_i E(n_{i-1},k_{i-1}) \end{aligned}$$

holds in order of magnitude. Equivalently we have

$$E(n_i,k_i) = \sqrt{\frac{n_i}{n_{i-1}k_i}}\Big(\frac{n_{i-1}}{n}\Big)^{\frac{\alpha}{2}} + E(n_{i-1},k_{i-1}). \tag{39}$$

Considering the network division (16) and the factor $k_i = \Omega(1)$ for all layer, we obtain

$$E(n_i,k_i) = n^{\frac{(i-h-1)\alpha+1}{2h-1}} k^{-\frac{2+2i\alpha-3\alpha}{4h-2}} + E(n_{i-1},k_{i-1}). \tag{40}$$

For $1 \le i \le h-1$, summing (40) up, we have

$$E(n_{h-1},k_{h-1}) = \sum_{i=2}^{h-1} n^{\frac{(i-h-1)\alpha+1}{2h-1}} k^{-\frac{2+2i\alpha-3\alpha}{4h-2}} + E(n_1,k_1).$$

$E(n_1,k_1) = \big(\sqrt{\frac{n_1}{n}}\big)^\alpha = n^{\frac{(1-h)\alpha}{2h-1}} k^{-\frac{\alpha}{4h-2}}$, which is smaller than the first term on the right-hand side in order of magnitude. Thus, the power consumed at the $(h-1)$th layer is

$$E(n_{h-1},k_{h-1}) = O\Big(n^{\frac{-2\alpha+1}{2h-1}} k^{-\frac{2h\alpha-5\alpha+2}{4h-2}}\Big). \tag{41}$$

For $i = h$ in (40), substituting $E(n_{h-1},k_{h-1})$ with (41), we can obtain the final result

$$E(n,k) = O\Big(n^{\frac{1-\alpha}{2h-1}} k^{-\frac{2h\alpha-3\alpha+2}{4h-2}}\Big). \tag{42}$$

2) Energy Consumption of the CMMM: Our CMMM strategy consumes the same amount of energy to transmit a bit as that of the MMM strategy, i.e., (42) also holds for the CMMM, which is supported by the following reasons.

- The network division is identical in two strategies.
- In two strategies, we build the same number of MTs at each layer, which leads to the same amount of power to transmit one bit.

3) Energy Consumption of the DMM and the CDMM: Intuitively, the DMM and CDMM use direct MIMO transmission, which is less energy-efficient than multihop MIMO transmission. At the top layer, to transmit n_{h-1} bits to all $k_{c_h} = \Theta(k)$ destination clusters, nodes in the source cluster broadcast data among the whole network. Thus, the energy to transmit one bit to all $\Theta(k)$ destination clusters is $O(1)$ on average. The result is identical in two strategies

$$E(n,k) = O\Big(\frac{1}{k}\Big). \tag{43}$$

VI. Discussion

A. Advantage of Cooperation

In our cooperative multicast scheme, we assume that the nodes nearby help each other on transmitting and receiving. Moreover, the hierarchical scheme proposed can bring about great improvements in the throughput only when h is sufficiently large. When setting h to 1, we cannot obtain a good capacity result since the cooperation is not fully utilized in this case. We know that the cooperation between nodes becomes stronger as h increases. In such case, we get a gain in the achievable throughput compared to [25] and [26]—particularly, a gain of $\Theta\big(\sqrt{\frac{n}{k}}\big)$ compared to [26]. The reason for the improvement is that when using distributed MIMO transmission, we exploit interference cancellation and enable simultaneous transmission of many bits. This method reduces the average interference level caused by each multicast session, which is the bottleneck of the achievable throughput.

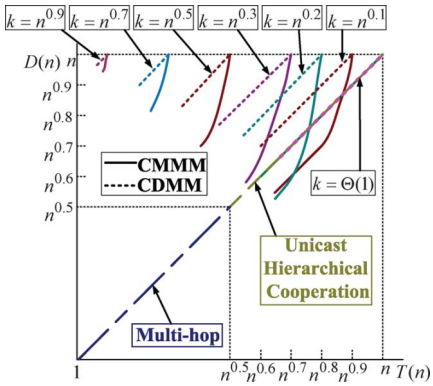


Fig. 2. Throughput-delay tradeoff for the CMMM and the CDMM compared to known results. The upper-right part of curves is achieved when choosing larger k. When $k = \Theta(1)$, the CMMM line overlaps the CDMM one, while the starting points are different. For two curves with the same k, we use a common color.

B. Effect of Different Network Divisions

Although we use cooperative schemes, it is still possible that the throughput cannot be improved. An obvious example is broadcast, where the number of clusters at each layer is smaller than that of the destination nodes, i.e., $n_{c_i} = O(k_i)$ for $2 \le i \le h$. Under such network division, even if $k = O(n^{1-\epsilon})$ for a small $\epsilon > 0$, we still cannot achieve a gain in the throughput.

Assume that we partition the network as $n_{c_i} = O(k_i)$ at the ith layer, it follows that $k_{c_i} = \Theta(n_{c_i})$. The reason that we cannot improve the throughput lies in the number of multicast sessions m_i (or converge multicast frames): Since $m_{i-1} = \Theta(m_i k_{c_i})$, $m_{i-1} = \Theta\big(\frac{m_i n_i}{n_{i-1}}\big)$ is greater than m_i in order of magnitude, meaning that the transmission scale grows as the layer becomes lower, which counteracts the advantage of parallel communications at lower layers, and results in no gain in the achievable throughput.

Moreover, in the MMM and DMM strategies, the delay decreases as k increases. When performing multicast, we need to transmit $B_h = \prod_{i=1}^{h-1} n_i$ bits to other cooperative nodes to prepare for distributed MIMO, which is also determined by the network division. The time cost on distributing B_h bits is the deterministic factor of delay and is reduced as k grows.

C. Delay-Throughput Tradeoff

First of all, we discuss how the number of destination nodes k affects the delay-throughput tradeoff. The delay-throughput tradeoff $D(n,k)/T(n,k)$ for multicast traffic is approximately $D/T = \widetilde{\Theta}(k)$, which is identical to that under noncooperative schemes. As shown in Fig. 2, when k grows, the tradeoff curves of CMMM/CDMM move leftwards, indicating the increase of D/T. The reason is obvious: When k increases, each source node has to deliver more copies of data within the network. Thus, the time to complete a multicast session gets longer, and D/T become larger.

However, exceptions exist: The CMMM curves for $k = 1$ and $k = n^{0.2}$ intersect as shown in the figure, which means that multicast D/T may be better than that of unicast for certain h. This is due to the existence of k_{th} in (23): In the CMMM strategy, when $k < k_{\text{th}} = \widetilde{\Theta}(n^{\frac{1}{2h}})$, Assumption 3 cannot be ensured at the second layer, i.e., $l_2 = k(\frac{n}{k})^{\frac{1}{2h-1}} \log^{h-3}(\frac{n}{k}) = O\big((\frac{n}{k})^{\frac{2}{2h-1}}\big)$. Thus, l_1 can only be derived by Lemma 4.7(b): $l_1 = O(\log(\frac{n}{k}))$. However, when $k > k_{\text{th}}$ and Assumption 3 holds, l_1 can be expressed as $l_1 = \Theta\big(n^{-\frac{1}{2h}} k^{\frac{2h}{2h-1}} \log^{h-2}(\frac{n}{k})\big)$. Combining the relationship between l_i and m_i, we obtain the number of transmission frames at the bottom layer shown in (24), and we repeat it here

$$m_1 = \begin{cases} (\frac{n}{k})^{\frac{2h-4}{2h-1}} \log(\frac{n}{k}), & \text{when } k = O(k_{\text{th}}) \\ k(\frac{n}{k})^{\frac{2h-5}{2h-1}} \log^{h-2}(\frac{n}{k}), & \text{when } k = \Omega(k_{\text{th}}). \end{cases}$$

In unicast, $k = 1$ is always below the threshold k_{th}; accordingly, the number of frames at the bottom layer can only be upper-bounded by $m_1 = \widetilde{\Theta}\big(n^{\frac{2h-4}{2h-1}}\big)$. However, $k = n^{0.2} > k_{\text{th}}$ when the number of layers $h > 2.5$, and therefore we can bound m_1 with $m_1 = \widetilde{\Theta}\big(k(\frac{n}{k})^{\frac{2h-5}{2h-1}} \log^{h-2}(\frac{n}{k})\big)$. If we choose $h = 3$, then $m_1 = \widetilde{\Theta}(n^{0.4})$ when $k = 1$; $m_i = \widetilde{\Theta}(n^{0.36})$ when $k = n^{0.2}$. Hence, in this case, the number of frames at the bottom layer of multicast is smaller than that of unicast. By the conclusion of Remark 4.2, the number of frames at the bottom layer will dominate the transmission time of each round, which results in a larger D/T of unicast case.

The effect of k_{th} is also embodied in Fig. 2. Since the number of destination nodes k is smaller than the threshold $k_{\text{th}} = \widetilde{\Theta}(n^{\frac{1}{2h}})$ only when k and h are both small, a typical example is $k = n^{0.1}$, shown in Fig. 2. The lower-left part of it is a straight line, indicating $h \le 5$ and $k < k_{\text{th}}$. In this case, the delay-throughput ratio D/T can only be lower-bounded by $\Theta(k^{-1})$. However, when $h \ge 5$, $k > k_{\text{th}}$, D/T is bounded by $\Theta\big(k(\frac{n}{k})^{-\frac{2}{2h-1}}\big)$, which is indicated by the upper-right part of the curve. As for other CMMM curves, the number of destination nodes k is never below the threshold since $h > 2$ in the CMMM strategy. Thus, the threshold has no effect on them.

Second, when considering the tradeoff $D(n,k)/T(n,k)$, the CMMM has a better performance. However, this tradeoff becomes worse as the number of layers h grows, which is also shown in Fig. 2. In fact, the delay is the time to complete a round in the CMMM, and for each round, only a certain number of $n \times \frac{n_{h-2}}{n_{h-1}}$ nodes act as source nodes. When the number increases, the time to finish a round will also increase. Nevertheless, this does not affect the multicast throughput since the number of bits transmitted in a round is linear to the time cost of a round. Hence, the tradeoff ratio D/T increases when the transmission scale of each round grows. Particularly, if all n nodes would act as source nodes in a round, the tradeoff $D/T = k$ is independent of h. In our scheme, however, there are $n \times (\frac{n}{k})^{\frac{2}{2h-1}}$ active nodes each round, and the transmission scale grows as h increases, which results in the phenomena above.

Another interesting phenomenon worth noting is that the delay-throughput tradeoff results are poor in the MMM and DMM strategies in accordance with the results in Section V-A, but the tradeoff ratio D/T is surprisingly identical to that of the CMMM and CDMM when $k = \Theta(n)$. In other words, the tradeoff results of the four strategies are unified to $D/T = n$ in the broadcast case. To explain this, we explore the common features of the four strategies in the broadcast case: 1) the network division is the same (in broadcast, we only divide each layer into a constant number of clusters); 2) we schedule the transmissions at the bottom layer in the same way. The direct consequences of these features are: a) the size of packets that need to be distributed in Step 1 is the same ($\Theta(n)$ bits); b) the

time spent on MIMO transmission at each layer is $\widetilde{\Theta}(1)$ for each source cluster; and c) the identical transmission strategy at the bottom layer results in the same amount of transmission time. Thus, for the four strategies, the throughput and delay are both identical in the broadcast case, leading to the unification of tradeoff ratio.

D. Multihop Versus Direct MIMO Transmission

For a given h, the throughput and delay of the MMM/CMMM are both better than that of the DMM/CDMM. Two factors contribute to the smaller delay: 1) parallel MIMO transmissions (the average time to complete the transmission of a MT at layer i is $O(\sqrt{n_{i-1}k_{c_i}/n_i})$, which is smaller than that of direct transmission, namely one slot); 2) less bits are transmitted at each round in the CMMM. By reducing the transmission time, the multihop scheme also improves the throughput. Comparing (15) to (28), the throughput gain is $(\frac{n}{k})^{\frac{h-1}{h(2h-1)}}$. Thus, the delay-throughput tradeoff of the CMMM is better than that of the CDMM, which is shown in Fig. 2.

The energy consumption in the multihop is approximately $k^{\frac{\alpha-2}{2}}$ times smaller than that in the direct MIMO transmission. This is because the multihop way that performs several short distance communications is more energy efficient than the direct manner.

E. MMM Versus Existing Approaches in the Literature

Now, we compare the MMM scheme to some other existing schemes published previously. We compare the MMM to cooperative schemes proposed in [22], [24], [25], and [32]. In [22], [24], and [25], the authors study multicast capacity under the protocol model in static networks. In [22] and [24], the authors establish the routing by constructing a Euclidean tree for multicast. Information is then transmitted from the source to destinations through the constructed tree. In [32], the authors study the throughput and the delay for multicast in mobile networks. They propose several approaches for multicasting transmission such as 2-hop relay with/without redundancy and multihop relay with redundancy. In [25], the authors consider multicast capacity in a more realistic and less pessimistic channel models. They propose a multicast routing and time scheduling scheme to achieve the computed asymptotic bound over all channel models except the simple Protocol Model.

The throughput comparison is listed in Table I. From the table, we can see that the MMM achieves much larger throughput than the scheme proposed in [22], [24], and [25]. In [22], [24], and [25], a large number of extra transmissions are wasted for redundancy in the routing process. Moreover, all the adjacent transmission has to be treated as interference while it is efficiently canceled in our MMM scheme. These two factors lead to the inferior throughput performance in [22], [24], and [25]. Compared to the three relay schemes proposed in [32], our MMM scheme also can guarantee a good aggregate throughput, which is close to the upper bound with a difference of only $\log n$ factor. This is almost the same as the result achieved in the 2-hop relay without redundancy in [32], $\Theta\left(\frac{n}{k}\right)$. Moreover, the authors also study multicast capacity in mobile networks under a more realistic channel model in [25], and they achieve the same capacity result $\Theta\left(\frac{n}{k}\right)$, which is also included in the result under the MMM scheme when h goes to infinity.

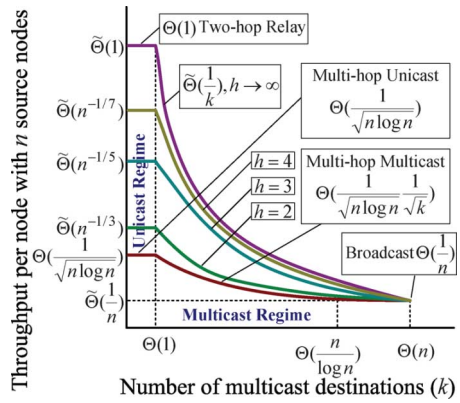


Fig. 3. We compare the known throughput results in static and mobile networks to that of our MMM strategy when $h = 2, 3, 4$. It shows that the MMM strategy can achieve a higher throughput than that of noncooperative schemes and the information-theoretic upper bound up to a logarithmic term when $h \to \infty$.

TABLE I
COMPARISON ON THROUGHPUT BETWEEN OUR MMM SCHEME AND SOME APPROACHES PUBLISHED PREVIOUSLY

schemes (static)	aggregate throughputs
MMM	$\Theta\left(\left(\frac{n}{k}\right)^{\frac{2h-2}{2h-1}}\log^{-(h-2)}\frac{n}{k}\right)$
multi-hop relay in [22]	$\Theta\left(\frac{n}{\sqrt{nk\log n}}\right)$
spanning routing tree in [24]	$\Theta\left(\frac{n}{k\log n}\right)$ $(k=O\left(\frac{n}{\log n}\right))$ $\Theta(1)$ $(k=\Omega\left(\frac{n}{\log n}\right))$
Multi-hop scheme in [25]	$\Theta\left(\sqrt{\frac{n}{k}}\right)$ $(k\le\frac{n}{\log^3 n})$ $\Omega\left(\frac{n}{k\sqrt{\log^3 n}}\right)$ $(\frac{n}{\log^3 n}\le k\le\frac{n}{\log^2 n})$ $\Omega\left(\sqrt{\frac{n}{k\log n}}\right)$ $(\frac{n}{\log^2 n}\le k\le\frac{n}{\log n})$ $\Theta(1)$ $(k\ge\frac{n}{\log n})$
scheme (mobile)	**aggregate throughput**
routing scheme in [25]	$\Theta\left(\frac{n}{k}\right)$
2-hop relay in [30]	$\Theta\left(\frac{n}{k}\right)$
2-hop relay in [30]	$\Omega\left(\frac{n}{\sqrt{n\log k}}\right)$
multi-hop relay in [30]	$\Omega\left(\frac{1}{\log n}\right)$

To better demonstrate the gain achieved in our MMM scheme, we also illustrate the throughput performance in Fig. 3, comparing with other known results. It can be seen that for any $\epsilon > 0$, our cooperative scheme obtains a throughput of $\Omega\left(\left(\frac{n}{k}\right)^{1-\epsilon}\right)$, with sufficiently large h. However, the delay performance of the MMM strategy is poor. This is because each node must transmit a large amount of bits at one time to achieve this throughput. Hence, if the delay performance is preferred, our MMM scheme may not be the appropriate choice.

VII. CONCLUSION

In this paper, we have developed a class of hierarchical cooperative schemes achieving an aggregate throughput of $\Omega\left(\left(\frac{n}{k}\right)^{1-\epsilon}\right)$ for any $\epsilon > 0$, which is arbitrarily close to the upper bound. Our proposed schemes rely on MIMO transmissions and consist of three steps. In Steps 1 Step 3, we use multilayer solutions to communicate within the clusters to maximize the aggregate throughput. We have analyzed the delay and energy consumption in each strategy and found

that the converge-based multihop scheme performs better in terms of both throughput and delay. Moreover, our CMMM strategy achieves the delay-throughput tradeoff identical to that of noncooperative schemes when $h \to \infty$. While for certain k and h, the tradeoff ratio can be even better than that of unicast.

REFERENCES

[1] 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.

[2] A. Özgür and O. Lévêque, "Throughput-delay trade-off for hierarchical cooperation in ad hoc wireless networks," presented at the Int. Conf. Telecommun., Jun. 2008.

[3] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[4] M. Franceschetti, O. Dousse, D. 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.

[5] 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.

[6] S. Aeron and V. Saligrama, "Wireless ad hoc networks: Strategies and scaling laws for the fixed snr regime," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2044–2059, Jun. 2007.

[7] J. Ghaderi, L. Xie, and X. Shen, "Throughput optimization for hierarchical cooperation in ad hoc networks," in *Proc. IEEE ICC*, May 2008, pp. 2159–2163.

[8] S. Vakil and B. Liang, "Effect of joint cooperation and multi-hopping on the capacity of wireless networks," in *Proc. IEEE SECON*, Jun. 2008, pp. 100–108.

[9] U. Niesen, P. Gupta, and D. Shah, "On capacity scaling in arbitrary wireless networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 9, pp. 3959–3982, Sep. 2009.

[10] M. J. Neely and E. Modiano, "Capaicty and delay tradeoffs for ad hoc mobile networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1917–1937, Jun. 2005.

[11] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," in *Proc. IEEE INFOCOM*, Mar. 2004, vol. 1, pp. 464–475.

[12] X. Lin and N. B. Shroff, "The fundamental capacity-delay tradeoff in large mobile wireless networks," Tech. Rep., 2004 [Online]. Available: http://cobweb.ecn.purdue.edu/~linx/papers.html

[13] A. Agarwal and P. Kumar, "Capacity bounds for ad hoc hybrid wireless networks," *Comput. Commun. Rev.*, vol. 34, no. 3, pp. 71–81, Jul. 2004.

[14] U. Kozat and L. Tassiulas, "Throughput capacity of random ad hoc networks with infrastructure support," in *Proc. ACM MobiCom*, Jun. 2003, pp. 55–65.

[15] B. Liu, Z. Liu, and D. Towsley, "On the capacity of hybrid wireless networks," in *Proc. IEEE INFOCOM*, 2003, vol. 2, pp. 1543–1552.

[16] B. Liu, P. Thiran, and D. Towsley, "Capacity of a wireless ad hoc network with infrastructure," in *Proc. ACM MobiHoc*, Sep. 2007, pp. 239–246.

[17] P. Li, C. Zhang, and Y. Fang, "Capacity and delay of hybrid wireless broadband access networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 2, pp. 117–125, Feb. 2009.

[18] C. Zhang, Y. Fang, and X. Zhu, "Throughput-delay tradeoffs in large-scale MANETs with network coding," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 199–207.

[19] L. Ying, S. Yang, and R. Srikant, "Optimal delay-throughput trade-offs in mobile ad hoc networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 4119–4143, Sep. 2008.

[20] S. Toumpis, "Asymptotic capacity bounds for wireless networks with non-uniform traffic patterns," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2231–2242, Jun. 2008.

[21] A. Keshavarz-Haddad, V. Ribeiro, and R. Riedi, "Broadcast capacity in multihop wireless networks," in *Proc. ACM MobiCom*, Sep. 2006, pp. 239–250.

[22] Z. Wang, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves, "A unifying perspective on the capacity of wireless ad hoc networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 211–215.

[23] X. Li, "Multicast capacity of wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 950–961, Jun. 2008.

[24] B. Liu, D. Towsley, and A. Swami, "Data gathering capacity of large scale multihop wireless networks," in *Proc. IEEE MASS*, 2008, pp. 124–132.

[25] S. Li, Y. Liu, and X. Li, "Capacity of large scale wireless networks under Gaussian channel model," in *Proc. ACM MobiCom*, 2008, pp. 140–151.

[26] X. Li, S. Tang, and O. Frieder, "Multicast capacity for large scale wireless ad hoc networks," in *Proc. ACM MobiCom*, Sep. 2007, pp. 266–277.

[27] A. Keshavarz-Haddad and R. Riedi, "Multicast capacity of large homogeneous multihop wireless networks," in *Proc. WiOPT*, Apr. 2008, pp. 116–124.

[28] P. Jacquet and G. Rodolakis, "Multicast scaling properties in massively dense ad hoc networks," in *Proc. ICPADS*, Jul. 2005, pp. 93–99.

[29] S. Shakkottai, X. Liu, and R. Srikant, "The multicast capacity of large multihop wireless networks," in *Proc. ACM MobiHoc*, Sep. 2007, pp. 247–255.

[30] Z. Wang, S. Karande, H. R. Sadjadpour, and J. J. Garcia-Luna-Aceves, "On the capacity improvement of multicast traffic with network coding," in *Proc. IEEE MILCOM*, Sep. 2008, pp. 1–7.

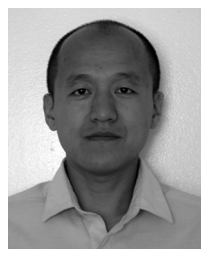
[31] U. Niesen, P. Gupta, and D. Shah, "The multicast capacity region of large wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1881–1889.

[32] C. Hu, X. Wang, and F. Wu, "MotionCast: On the capacity and delay tradeoffs," in *Proc. ACM MobiHoc*, May 2009, pp. 289–298.

[33] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, L. Mo, W. Dong, Z. Yang, M. Xi, J. Zhao, and X. Li, "Does wireless sensor network scale? A measurement study on GreenOrbs," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 10–15, 2011, pp. 873–881.

[34] X. Li, Y. Liu, S. Li, and S. Tang, "Multicast capacity of wireless ad hoc networks under Gaussian channel model," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1145–1157, Aug. 2010.

[35] C. Hu, X. Wang, D. Nie, and J. Zhao, "Multicast scaling laws with hierarchical cooperation," in *Proc. IEEE INFOCOM*, San Diego, CA, Mar. 2010, pp. 1–9.



Xinbing Wang (M'06) received the B.S. degree (with hons.) in automation from Shanghai Jiao Tong University, Shanghai, China, in 1998, the M.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2001, and the Ph.D. degree with a major in electrical and computer engineering and minor in mathematics from North Carolina State University, Raleigh, in 2006.

Currently, he is a faculty member with the Department of Electronic Engineering, Shanghai Jiao Tong University. His research interests include resource allocation and management in mobile and wireless networks, TCP asymptotics analysis, wireless capacity, cross-layer call admission control, asymptotics analysis of hybrid systems, and congestion control over wireless ad hoc and sensor networks.

Dr. Wang has been a member of the Technical Program Committees of several conferences including IEEE INFOCOM 2009–2011, IEEE ICC 2007–2011, and IEEE GLOBECOM 2007–2011.

Luoyi Fu received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009, and is currently working with Prof. Xinbing Wang toward the Ph.D. degree in electronic engineering at the same university.

Her research of interests are in the area of scaling laws analysis in wireless networks.

Chenhui Hu received the B.S. and M.S. degrees in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2010, respectively, and is currently pursuing the Ph.D. degree at Harvard University, Cambridge, MA.

From 2007 to 2010, he was doing research at the Institute of Wireless Communication Technology (IWCT), Shanghai Jiao Tong University, supervised by Prof. Xinbing Wang and Youyun Xu. His research interests include wireless capacity and connectivity, asymptotic analysis of mobile ad hoc networks, multicast, distributed MIMO, and percolation theory.