

Converge-Cast with MIMO

Luoyi Fu, Yi Qin, Xinbing Wang
 Dept. of Electronic Engineering
 Shanghai Jiao Tong University, China
 Email: {yiluofu,qinyi_33,xwang8}@sjtu.edu.cn

Xue Liu
 Dept. of Computer Science & Engineering
 University of Nebraska Lincoln, USA
 Email: xueliu@cse.unl.edu

Abstract—This paper investigates throughput and delay based on a newly predominant traffic pattern, called converge-cast, where each of the n nodes in the network act as a destination with k randomly chosen sources corresponding to it. Adopting Multiple-Input-Multiple-Output (MIMO) technology, we devise two many-to-one cooperative schemes under converge-cast for both *static* and *mobile* ad hoc networks (MANETs), respectively. In a static network, our scheme highly utilizes hierarchical cooperation MIMO transmission. This feature overcomes the bottleneck which hinders converge-cast traffic from yielding ideal performance in traditional ad hoc network, by turning the originally interfering signals into interference-resistant ones. It helps to achieve an aggregate throughput up to $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$. In the mobile ad hoc case, our scheme characterizes on *joint* transmission from *multiple* nodes to *multiple* receivers. With optimal network division where the number of nodes per cell is constant bounded, the achievable per-node throughput can reach $\Theta(1)$ with the corresponding delay reduced to $\Theta(k)$. The gain comes from the strong and intelligent cooperation between nodes in our scheme, along with the maximum number of concurrent active cells and the shortest waiting time before transmission for each node within a cell. This, to a great extent, increases the chances for each destination to receive the data it needs with minimum overhead on extra transmission. Moreover, our converge-based analysis well unifies and generalizes previous work since the results derived from converge-cast in our schemes can also cover other traffic patterns. Last but not least, our cooperative schemes are of interest not only from a theoretical perspective but also shed light on future design of MIMO schemes in wireless networks.

I. INTRODUCTION AND RELATED WORK

Fueled by the seminal work of Kumar [1] *et al.*, who showed that the optimal static unicast capacity is $\Theta(\frac{1}{\sqrt{n}})$ and $\Theta(\frac{1}{\sqrt{n \log n}})$ ¹ for random network, capacity analysis of ad hoc networks have triggered great interest. Later on, Grossglauser and Tse [2] demonstrated that $\Theta(1)$ capacity per source-destination (S-D) pair is achievable if taking mobility of the network into account, but packets have to endure a larger delay. Due to the phenomenon that larger capacity is at the cost of a larger delay, some analysis on capacity-delay tradeoffs arises. One interesting work is from Neely and Modiano

[3] who introduced redundant packets transmission through multiple opportunistic paths to reduce delay while a decrease on capacity is also incurred. Under *i.i.d.* mobility, the per-node capacity is shown to be $T(n) = \Theta(1)$ and delay $D(n)$ yielded to scale as $\Theta(n \cdot T(n))$ [3]. Later work also studied the tradeoff between capacity and delay, where nodes either perform traditional operations such as storage, replication and forwarding ([4]- [5]) or transmit through coding or infrastructure support ([6]- [8]).

However, all the results above strongly rely on the assumption that all the concurrent transmissions are always interfering with others. This becomes a limitation which largely constrains the capacity. In contrast, MIMO enables nodes to perform cooperative communication by turning mutually interfering signals into useful ones, where the gain of capacity can then be obtained. The gain is well demonstrated by Aeron *et al.* [9] who presented a MIMO collaborative strategy which achieves a per-node capacity of $\Theta(n^{-1/3})$. Following that, Özgür *et al.* [14] constructed a hierarchical cooperative scheme relying on distributed MIMO communications to achieve a linear capacity scaling. It turns out that nearly all the interferences can be canceled through hierarchical cooperation. Thereon, multicast scaling is taken into account in [13] under hierarchical cooperation which achieves an aggregate capacity of $\Omega\left(\left(\frac{n}{k}\right)^{1-\epsilon}\right)$ for any $\epsilon > 0$. This also achieves a gain on capacity compared with previous works on multicast such as [10]- [12].

While the tradeoff for *unicast* and *multicast* traffic pattern have been extensively studied in previous work, *converge-cast* is still a relatively new concept and under active research. *Converge-cast* refers to a communication pattern in which the flow of data from a set of nodes transmit to a single node, either directly or over multi-hop routes. Recently, there appeared many new applications such as real-time multimedia, battlefield communications and rescue operations that impose stringent capacity-delay requirements on converge-cast.

In this paper, we jointly consider the effect of converge-cast and cooperative strategies on asymptotic performance of networks. The motivations come from the following reasons: **1.** Although there have been some researches on converge-cast (such as [15], [16], [17], [18]), their major concern is limited to the extreme case where all nodes flow data to a single sink in the network. However, a wide range of applications such as machine failure diagnosis, pollutant detection and supply chain

¹We use the following notation throughout our paper:

$f(n) = o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$, $f(n) = \omega(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0$,
 $f(n) = O(g(n)) \Leftrightarrow \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty$, $f(n) = \Omega(g(n)) \Leftrightarrow \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)} > 0$, $f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n))$ and $g(n) = O(f(n))$, $f(n) = \tilde{\Theta}(\cdot)$: The corresponding order $\Theta(\cdot)$ which contains a logarithmic order.

management may require multiple such converge-cast groups existing in parallel in the network rather than a single one. **2.** Unlike multicast where the transmission process becomes more and more diverse, vast space of further improvement on its performance can be discovered in converge-cast, due to its convergent process. **3.** Since distinctive sources may transmit different data to their common destination, such traffic pattern can be treated as a generalized reversed “multicast”. This ensures a coverage on other kinds of traffic modes such as unicast, multicast and broadcast, since all of them can be regarded as special cases of converge-cast. To our best knowledge, there are no previous study on the network performance under converge-cast with MIMO.

Concentrating on throughput and delay performance in this paper, we propose a new type of many-to-one cooperative schemes with MIMO in both static and mobile networks, from the perspective of converge-cast. First, we design a many-to-one cooperative scheme in a static network, where the whole network is divided into clusters with equal number of nodes in each of them. Communications between clusters are conducted through distributed MIMO transmissions combined with multi-hop strategy while within a cluster it is operated through joint transmission of multiple nodes to others once a time. Each cluster can be treated as a subnetwork and further divided into smaller clusters. This process is carried out through hierarchical operation. The multiple-transmission-multiple-reception feature of MIMO suits the many-to-one characteristic of converge-cast well. In a traditional ad hoc network, only one transmission can be active at a time while all the adjacent transmissions are treated as interference. This imposes a significant bottleneck on converge-cast and makes it impossible to achieve ideal performance. However, this bottleneck can be removed with the adoption of MIMO. The gain comes from the smart transformation from interfering signals into useful ones to the receivers through hierarchical cooperative transmissions in the scheme.

Under MANETs where hierarchical cooperation cannot be established due to the mobility of nodes, we devise another many-to-one cooperative scheme where the network is still divided into equal cells. In each time slot, multiple nodes that possess information for the same destination are allowed for joint transmission to other nodes within the cell. Other nodes will receive a combination of the information from these transmitters due to the effect of MIMO through fading channels. This procedure continues, with the number of nodes that hold such mixed information increases, until all the destinations receive sufficient mixed information that can be decoded with high probability.

Our main contributions can be summarized as follows:

- Our many-to-one cooperative scheme in a static network breaks the bottleneck hindering converge-cast from achieving ideal performance in a traditional network by converting adjacent interfering signals into useful ones. The achievable aggregate throughput can be up to $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$, which nearly approaches the upper-bound.

- For our many-to-one cooperative scheme under MANETs, the optimal choice for network division is constant-bounded number of nodes per cell. When combined with MIMO, this allows the maximum number of concurrent transmitting cells as well as the shortest waiting time before a transmission for each node within a cell. This leads to a per-node throughput of $\Theta(1)$ with the corresponding delay reduced to $\Theta(k)$.
- Our results well unify and generalize the previous work (such as [6] in MANET and [14], [15]- [18] in static network) since all of them can be easily applied to other traffic modes. Furthermore, our novel many-to-one cooperative schemes provide useful guidelines for future design of MIMO schemes in wireless networks. Especially, our scheme in MANETs breaks the vacancy of such MIMO scheme design remaining in mobile networks before.

The rest of the paper is organized as follows. In Section 2, we present the models and definitions. In Section 3 and Section 4, we describe our cooperative schemes under static and mobile ad hoc networks, respectively. The corresponding throughput and delay achieved based on the two schemes are also presented in detail in these two sections. All the results are further discussed in Section 5. Finally, we present concluding remarks in Section 7. Some proofs are provided in-line and others are in Appendix.

II. MODELS AND DEFINITIONS

A. Network Model

In this paper, we consider an ad hoc network where nodes are randomly positioned in a unit square.

Traffic Pattern: In converge-cast scenario, we assume n nodes located in the network with each one serving as a destination. For each destination node, there are k randomly and independently chosen sources. These k nodes then send packets to their common destination. In multicast, all the packets sent out from a source node are the same while in convergecast, the packets from those k sources may be totally different and all of them are indispensable to form the complete information. Moreover, the data rates of each edge of the spanning tree in multicast are all same while they are different in each edge in convergecast.

Physical Layer model: We assume that communication takes place over a channel with limited bandwidth W . Each node has a power budget P . The channel gain between two nodes v_i and v_j at time t is given by:

$$h_{ij}[t] = \sqrt{G}d_{ij}^{-\alpha/2}e^{j\theta_{ij}[t]}, \quad (1)$$

where d_{ij} is the distance between the nodes, $\theta_{ij}[t]$ is the random phase at time t , uniformly distributed in $[0, 2\pi)$. $\{\theta_{ik}[t]\}$ are *i.i.d.* random processes across all i and k , independent of each other. G and the path loss propagation $\alpha \geq 2$ are assumed to be constants. Then, the signal received by node i at time t

can be expressed as

$$Y_i[t] = \sum_{j \in \mathbb{T}[t]} h_{ij}[t] X_j[t] + Z_i[t] + I_i[t], \quad (2)$$

where $Y_i[t]$ is the signal received by node v_i at time t , $\mathbb{T}[t]$ represents the set of active senders transmitting signals to v_i , which can be added constructively, $Z_i[t]$ is the additive white Gaussian noise at v_i with variance N_0 per symbol and $I_i[t]$ is the interference from the nodes. Moreover, we assume each node is equipped with one antenna.

B. Definitions

Converge-cast Session: a converge-cast session is defined as the set composed of one destination and all its corresponding sources.

Delay: Delay is defined as the time a destination takes to receive all the packets from its corresponding k sources. The averaging is over all bits (or packets) transmitted in the network.

Throughput: A throughput $\lambda > 0$ is said to be feasible if each sources in a converge-cast session can send at a rate of at least λ bits per second to their common destination. Denoting $m(t)$ as the number of packets from sources that a destination receives in t time slots. Then, the long term per-node throughput is defined as

$$\lambda = \liminf_{t \rightarrow \infty} \frac{m(t)}{t}.$$

And the aggregate throughput is $\Lambda = n\lambda$.

C. Notations

In table I, we list all the parameters that will be used in later analysis, proofs and discussions.

III. MANY-TO-ONE SCHEME UNDER STATIC NETWORKS

In this section, we first present a many-to-one cooperative scheme with MIMO under static networks. Then we derive the information-theoretic upper bound of the converge-cast throughput. Finally, we analyze the throughput and delay achieved under our proposed scheme. Our results demonstrate that the achievable throughput can nearly approach the upper bound.

A. Many-to-one Cooperative Scheme 1 under Static Networks

As is shown in [14], hierarchical cooperation can achieve better throughput scaling than classical multihop schemes under certain assumptions on the channel model in static wireless networks. This motivates us to design a hierarchical scheme which can be applied to converge-cast.

1) *Scheduling Algorithm:* Under hierarchical schemes, a network is divided into equal clusters. Each cluster is then treated as a *subnetwork* and we can further divide the subnetwork into smaller clusters. With recursion operation, the procedure goes on until the network is divided into h layers with the original network at the h th layer and the 1st layer at the bottom one. A scheduling algorithm can be designed on each subnetwork at each layer. The algorithm keeps executing

TABLE I: Notations

Notation	Definition
n	The total number of nodes in the network.
k	The number of sources for each destination in the network.
h	The number of layers a network is divided into.
i	The i th layer of the network, where $1 \leq i \leq h$.
n_i	The number of nodes in the i th layer.
k_i	The number of sources for each destination node in the i th layer.
n_{c_i}	The number of clusters in the i th layer.
k_{c_i}	The number of source clusters in the i th layer.
t_i	The number of converge-cast sessions in the i th layer.
T_i	The aggregate throughput at layer i in static network.
$D(i, k)$	The average delay to complete a converge-cast session for a destination at layer i .
B_i	The minimum amount of data a node needs to send at layer i . We can also call it bulk size.
M	The average number of nodes in each cell.

from layer to layer, the process of which is similar per layer per cluster but with a larger scale as the number of layer i increase from 1 to h . The procedure continues until all the layers have finished the algorithm.

Since the algorithm is similar at each layer but with different scale, we will present our recursive cooperative scheme 1 at a particular layer i . The scheme is divided into three steps, which are described as follows:

Step 1. Preparing for Cooperation with Recursion: Since there are k_i source nodes belonging to one session at layer i , under converge-cast, they must distribute their packets² to some other nodes in the same cluster. For each node in the cluster, the k_{i-1} sources jointly transmit their packets to $\frac{n_{i-1}k_{i-1}}{k_i}$ nodes in the cluster, which receives a linear combination of that packet mixed with channel coefficients (including amplitude and phase information). The process keeps until all the $\frac{n_{i-1}k_{i-1}}{k_i}$ nodes except for these k_{i-1} sources receive the packets from them. Note that as for each transmission from the k_{i-1} sources to a specific node, the process is many-to-one transmission and this is equivalent to dividing the current cluster into smaller-size clusters and the similar procedure executes in a smaller cluster. This smaller cluster is treated as a new network where there are n_{i-1} nodes, with k_{i-1} sources for each destination. And the three-step algorithm can still be executed in this smaller network. Note that our algorithm starts from the bottom layer, *i.e.*, layer 1 of the network and continues to a higher layer until it reaches layer h .

Step 2. Multi-hop MIMO Transmissions: For the sake of

²Note that the size of packet in our cooperative scheme 1 differs at different layers. Each node divides a packet into $\frac{k_i-1}{k_i}$ packets. Each of these packets is then transmitted in the next layer $i-1$.

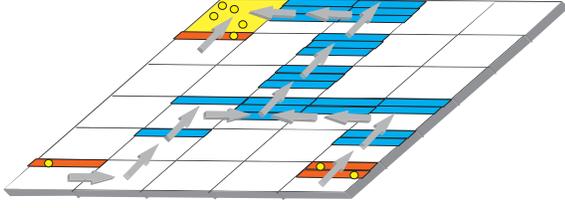


Fig. 1: An example of a CT in multi-hop MIMO transmission. Assume one destination has 4 sources. The red parts represent source clusters. 1/4 percent of the nodes are allowed for transmission in the two source clusters on the left while 1/2 percent of the nodes are active in the source cluster on the right. Whenever several clusters flow to a common cluster in the next time slot, that cluster will be colored with the part several times' larger than all the transmitting clusters to it. Only the destination cluster (colored with yellow) will be entirely colored when all the data finally flows to it.

energy efficiency, in this step, we construct a multi-hop routing mode rather than direct MIMO transmission between clusters. Several source clusters start a series of MIMO transmissions to reach their common destination clusters in multi-hop manner. Since each source cluster has $\frac{n_{i-1}k_{i-1}}{k_i}$ packets to send in one time slot, due to MIMO, several source clusters are allowed for concurrent transmission to one cluster at the same time slot. To achieve asymptotically optimal converge-cast capacity, we construct a converge-cast tree (CT) by conducting the three substeps presented below, spanning from source clusters S_{ij} s to their common destination clusters D_i . Here $1 \leq j \leq k_{c_i}$. Denote $P_i = \{S_{ij}, D_i, 1 \leq j \leq k_{c_i}\}$.

- 1) Constructing the Euclidean spanning tree \mathcal{E}_{EST} : Firstly we divide the square into cells with side length $\frac{1}{2^g}$, where $g = t-1$, $t = \lceil \log_4 k \rceil$. For each cell that contains $s \geq 2$ clusters in P_i , we randomly select a cluster p_{ij} . For any other $p_{ik} (k \neq j)$ in the cell, let $\mathcal{E}_{EST} \rightarrow \mathcal{E}_{EST} \cup \{\overline{p_{ix}p_{iy}}\}$ and $\mathcal{P} \rightarrow \mathcal{P} - \{p_{ik}\}$. Subsequently we conduct this process by letting $g = t-2, \dots, 1, 0$.
- 2) Getting the Manhattan routing tree \mathcal{E}_{MRT} : for each edge \overline{uv} in \mathcal{E} , assume that the coordinates of u and v are (i_u, j_u) and (i_v, j_v) , respectively. We then find a cluster w whose coordinate is (i_u, j_v) . Afterwards, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{\overline{uw}\}$, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{\overline{wv}\}$, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} - \{\overline{uv}\}$.
- 3) Obtaining the CT(P_i) for each edge \overline{uv} in \mathcal{E}_{MRT} , we connect clusters crossed by \overline{uv} in sequence to form a path, denoted as $E(u, w)$. Then $\mathcal{E}_{CT} \rightarrow \mathcal{E}_{CT} \cup E(u, w)$, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} - \{\overline{uv}\}$. Finally, \mathcal{E}_{CT} is the set of edges of CT(P_i).

Figure 1 shows a simple example of the data flow on such a converge-cast tree (CT).

Step 3. Cooperative Reception: Given the total number of converge-cast sessions t_i at layer i , consider a particular node in the cluster. It can receive $\frac{t_i k_i}{n_i}$ packets from other nodes, with each of them contributing $\frac{t_i k_i}{n_i n_{i-1}}$ packets. Considering n_{i-1} destinations in each cluster, the traffic load are $\Theta\left(\frac{t_i k_i n_{i-1}}{n_i}\right)$ packets. Since the data exchanges only involve intra-cluster communication, they can work according to 9-TDMA scheme

where the cells which are located 3 cells away from each other can be active concurrently.

2) *Throughput and Delay Analysis under Many-to-one Cooperative Scheme 1:* Now we focus on throughput and delay that can be achieved under the scheme presented in 3.1.1. We first present our main results as follows.

Theorem 1: In static wireless networks, by adopting our cooperative scheme 1, we can achieve an aggregate throughput of

$$\Lambda = \tilde{\Theta}\left(n^{\frac{2h-2}{2h-1}} \cdot k^{-\frac{1}{2h-1}}\right) \quad (3)$$

with the delay of

$$\mathbb{E}[T] = \begin{cases} \tilde{\Theta}\left(n^{\frac{2h^2-4h+3}{2h-1}} \cdot k^{-\frac{2h^2-2h-1}{2h-1}}\right), & \text{if } k = \Omega(n^{\frac{1}{2h-2}}) \\ \tilde{\Theta}\left(n^{\frac{h^2-2h+2}{2h-1}} k^{\frac{h^2-4h+3}{2h-1}}\right), & \text{if } k = O(n^{\frac{1}{2h-2}}) \end{cases} \quad (4)$$

To prove Theorem 1, we first introduce the following lemmas.

Lemma 1: Under converge-cast, with each of the n nodes in the network acting as destination and receiving packets from its distinctive k sources, the aggregate capacity is upper-bounded by

$$\sum_{i=1}^n \lambda_i \leq Cn \log n \quad (5)$$

where $C > 0$ is a constant independent of n .

Lemma 2: (Lemma 4.3 in [13]) By 9-TDMA scheme, when $\alpha > 2$, one node in each cluster has a chance to operate data exchanges at a constant transmission rate. Also when $\alpha > 2$, the interfering power received by a node from the simultaneously operating clusters is upper-bounded by a constant.

Lemma 3: Given k_i independently and uniformly distributed source nodes in the network at layer i , the number of source clusters k_{c_i} is given by

$$k_{c_i} = \begin{cases} \Theta(k_i), & \text{if } k_i = O(n_{c_i}) \\ \Theta\left(\frac{n_i}{n_{i-1}}\right), & \text{if } k_i = \Omega(n_{c_i}) \end{cases} \quad (6)$$

Proof: The proof is similar to that of Lemma 4.6 in [13] and we do not present the detailed proof here. ■

Lemma 4: When $t_i k_i = O((n_{c_i})^{p_2})$ holds for all layer i , where $2 \leq i \leq h$ and p_2 is a positive constant,

- if $k_i = \Omega(n_i \log n_{c_i})$, then $k_{i-1} = \Theta\left(\frac{k_i}{n_{c_i}}\right)$ w.h.p.
- if $k_i = O(n_i \log n_{c_i})$, then $k_{i-1} = O\left(\frac{k_i}{n_{c_i}}\right)$ w.h.p.

Proof: The proof is similar to that of Lemma 4.6 in [13] and we do not present the detailed proof here. ■

Consider the three steps in our scheme at layer i . Assume an aggregate converge-cast throughput $\tilde{\Theta}\left(n_{i-1}^a k_{i-1}^b\right)$ is achievable at layer $i-1$ w.h.p., where $0 \leq a \leq 1$, $-1 \leq b \leq 0$ and $a+b < 0$. It is easy to obtain that the total time to complete $k_i t_i$ traffic loads is

$$\tilde{\Theta}\left(\frac{k_i t_i n_{i-1}^{1-a}}{n_i k_{i-1}^b}\right) + O\left(k_i t_i \sqrt{\frac{k_i}{k_{i-1} n_i n_{i-1}}}\right) + \Theta\left(\frac{k_i t_i n_{i-1}^{1-a}}{n_i}\right). \quad (7)$$

Hence, the throughput can be expressed as

$$\begin{aligned} T_i &= \frac{k_i t_i}{\tilde{\Theta} \left(\frac{k_i t_i n_{i-1}^{1-a}}{n_i k_{i-1}^b} \right) + O \left(k_i t_i \sqrt{\frac{k_i}{k_{i-1} n_i n_{i-1}}} \right) + \Theta \left(\frac{k_i t_i n_{i-1}^{1-a}}{n_i} \right)} \\ &= \tilde{\Theta} \left(\frac{n_i n_{i-1}}{\sqrt{n_i n_{i-1}} + n_{i-1}^{2-a} k_{i-1}^{-b}} \right). \end{aligned} \quad (8)$$

In order to optimize the network division at layer i , we consider two cases, i.e., $n_{c_i} = O(k_i)$ and $n_{c_i} = \Omega(k_i)$. According to Lemma 4 and Lemma 5, we have the following two cases:

- 1) If $n_{c_i} = O(k_i)$, then $k_{c_i} = O(n_{c_i})$, $k_{i-1} = \tilde{\Theta} \left(\frac{k_i}{n_{c_i}} \right)$;
- 2) If $n_{c_i} = \Omega(k_i)$, then $k_{c_i} = \Theta(k_i)$, $k_{i-1} = \tilde{\Theta}(1)$;

In case 1, the throughput in Equation (8) can be written as

$$\begin{aligned} T_i &= \tilde{\Theta} \left(\frac{n_i n_{i-1}}{\sqrt{n_i n_{i-1} k_{c_i}} + n_{i-1}^{2-a} k_i^{-b} n_i^b n_{i-1}^{-b}} \right) \\ &= \tilde{\Theta} \left(\frac{1}{n_{i-1}^{-1} + n_{i-1}^{1-a-b} k_i^{-b} n_i^b} \right). \end{aligned} \quad (9)$$

The result is optimized when $n_{i-1} = \tilde{\Theta} \left(k_i^{\frac{-b}{a+b-2}} n_i^{\frac{b-1}{a+b-2}} \right)$.

Then, $n_{c_i} = \frac{n_i}{n_{i-1}} = k_i^{\frac{b}{a+b-2}} n_i^{\frac{a-1}{a+b-2}} = O(k_i)$ and $k_i = \Omega \left(n_i^{\frac{a-1}{a-2}} \right)$.

At the bottom layer, the aggregate throughput is $\frac{1}{k_1}$. If we divide the network in the optimal way at each layer, the relationship between n_i , k_i and throughput at each layer is $n_i = k_i^{\frac{b-1}{b}} n_{i-1}^{\frac{a+b-2}{b-1}}$ and $T_i = k_i^{\frac{-b}{a+b-2}} n_i^{\frac{b-1}{a+b-2}}$.

Note that $n_h = n$, we obtain the aggregate throughput at layer h , i.e.,

$$T = \tilde{\Theta} \left(n^{\frac{2h-2}{2h-1}} \cdot k^{-\frac{1}{2h-1}} \right). \quad (10)$$

In the optimized result, $k = \Omega \left(n^{\frac{a-1}{a-2}} \right) = \Omega \left(n^{\frac{1}{2h-2}} \right)$. Hence, $n_i = k_i^{\frac{1}{2i-2}} n_{i-1}^{\frac{2i-1}{2i-2}} \Rightarrow n_1 = n^{\frac{h}{2h-1}} k^{-\frac{h-1}{2h-1}}$.

Now we turn to the analysis on delay at layer i , denoted as $D(i, k)$. Through Equation (8), we get $D(h, k) = \frac{nB}{T}$, where B is minimum size of data transmitted at layer h , i.e., $B = \tilde{\Theta} \left(\prod_{i=1}^{h-1} \frac{n_i}{k_{i+1}} \right) = \tilde{\Theta} \left(\prod_{i=2}^{h-1} \frac{n_i}{k_i} \cdot \frac{n_1}{k} \right) = \tilde{\Theta} \left(\left(\frac{n}{k} \right)^{h-2} \frac{n_1}{k} \right) = (n)^{\frac{2h^2-4h+2}{2h-1}} k^{-\frac{2h^2-2h}{2h-1}}$.

And through recursion on $D(i, k)$, the final delay $D(h, k)$ can be obtained, i.e.,

$$D(h, k) = \tilde{\Theta} \left(n^{\frac{2h^2-4h+3}{2h-1}} k^{-\frac{2h^2-2h-1}{2h-1}} \right). \quad (11)$$

Then we focus on case 2, where $n_{c_i} = \Omega(k_i)$. Consider the aggregate throughput at layer i , we have

$$T_i = \tilde{\Theta} \left(\frac{n_i n_{i-1}}{\sqrt{n_i n_{i-1} k_i} + n_{i-1}^{2-a}} \right) = \tilde{\Theta} \left(\frac{n_i}{\sqrt{n_i k_i n_{i-1}^{-\frac{1}{2}}} + n_{i-1}^{1-a}} \right). \quad (12)$$

When the result is optimized, $n_{i-1} = \tilde{\Theta} \left((n_i k_i)^{\frac{1}{3-2a}} \right)$. Therefore, $n_{c_i} = \frac{n_i}{n_{i-1}} = k_i^{\frac{1}{2a-3}} n_i^{\frac{2a-2}{2a-3}} = \Omega(k_i)$ and $k_i = O \left(n_i^{\frac{a-1}{a-2}} \right) = O \left(n^{\frac{1}{2h-2}} \right)$.

Note that in this case, the aggregate throughput is 1 at the bottom layer since the traffic pattern can be treated as unicast at this layer. Following the same procedure in case 1, the aggregate throughput and delay $D(h, k)$ at layer h can be obtained, which are shown as follows:

$$T = T_h = \tilde{\Theta} \left(n^{\frac{2h-2}{2h-1}} k^{-\frac{1}{2h-1}} \right), \quad (13)$$

and

$$D(h, k) = \tilde{\Theta} \left(n^{\frac{h^2-2h+2}{2h-1}} k^{\frac{h^2-4h+3}{2h-1}} \right). \quad (14)$$

This completes our proof for THEOREM 1.

IV. MANY-TO-ONE SCHEME UNDER MANETS

In Section III, we analyze the performance in static networks. In this section, we turn to the mobile networks. Due to the mobility characteristics of nodes, the network performance may be quite different from that in static ones. In the following subsections, we will introduce the mobility model and present another scheme that is suitable for mobile networks. Then, we will give our analysis on throughput and delay obtained from the scheme.

A. Mobility Model

We introduce *two-dimensional i.i.d. mobility model* into the network, i.e., n nodes are uniformly distributed in the network. At the beginning of each time slot, each node randomly chooses a point in the unit square and moves there. In this model, we assume that the nodes move quickly so that the nodes' positions are independent from time slot to time slot. We also define it as *fast mobility model* where the mobility of nodes is at the same time-scale as that of data transmission.

B. Many-to-one Cooperative Scheme 2 under MANETs

When the position of nodes may be varying with time, it is impossible to construct a hierarchical scheme under mobile networks. Since the relationship determined in the current time slot between nodes may be destroyed in the next one due to the randomness incurred by mobility. Hence, we need to design a new scheme that can take advantage of mobility of the nodes. With appropriate scheduling, the network performance can be improved.

1) *Many-to-one Cooperative Scheme 2*: We divide the whole network into c cells such that there are M nodes in each cell on average. To avoid the interference incurred to the network from the neighboring cells, we adopt the 9-TDMA strategy illustrated in section III again.

According to our model definition, since each node acts as a destination, there are always some destinations in each cell.

- Each cell becomes active once every c_0 time slots. In an active cell, transmission occurs among the nodes within the same cell.

- In an active cell, in each time slot, if there exist both a destination and some of its sources, then we call there are *sources-destination pair* in the cell. If there are several such pairs in the cell, then we randomly choose one pair, and let all these sources in this pair form an antenna array (This is feasible under the assumption in footnote 4.) and jointly send their packets to their common destination as well as all the other nodes in that cell. All the other nodes except the destination pick out a stored packet which is for the same destination as the new one, linearly combine the new one with it and then replace the stored one in the relaying buffer.
- If there are no sources-destination pairs in the cell, choose the maximum number of sources that belong to the same destination in that cell. Then, the chosen sources jointly send their packets to all the other nodes in the same cell. Similarly, the nodes will linearly combine the new packet with the stored one and replace it in the relaying buffer.
- If there are neither sources-destination pairs nor sources that belong to the same destination in the cell, then choose the maximum number of relays which hold the packets that are to be transmitted to the same destination. Those chosen relays then jointly send their packets to all the other nodes in the same cell. After receiving the packet, all the node will update the packet as described above.

A simple illustration of our scheme is shown in Figure 2.

C. Analysis of Throughput and Delay under Many-to-One Scheme 2

In this subsection, we will analyze the achievable throughput and delay under our proposed scheme 2. First, we will first compute the bound of achievable delay and then analyze the corresponding throughput.

The main results obtained under scheme 2 is presented in the following theorem.

Theorem 2: Suppose $k = o(n)$, then under many-to-one cooperative scheme 2, with the optimal network division $M = \Theta(1)$, we can achieve ideal performance on both the average delay required for a destination to receive packets from all its k corresponding sources and the per-node throughput, listed as follows:

$$\begin{cases} \lambda = \Theta\left(\frac{1}{\log n}\right), \mathbb{E}[D_N] = \Theta(\log n) & \text{if } k = o(\log n) \\ \lambda = \Theta(1), \mathbb{E}[D_N] = \Theta(k) & \text{if } k = \omega(\log n) \end{cases}. \quad (15)$$

To prove Theorem 2, we turn to the proof for delay in IV-C1) first and then prove the throughput in IV-C2).

1) *Analysis on Delay:* Before the proof of delay, we first introduce the following two lemmas.

Lemma 5: Consider n nodes uniformly distributed in the network area. The network is divided into c identical cells. Then, the number of nodes in each cell is $M = \frac{n}{c}$ w.h.p. if

$$\lim_{n \rightarrow \infty} \frac{n}{c \log c} = \infty.$$

Lemma 6: As for a destination node, the condition that it can successfully decode the packets from all its k sources is that there should be at least k different linear combinations

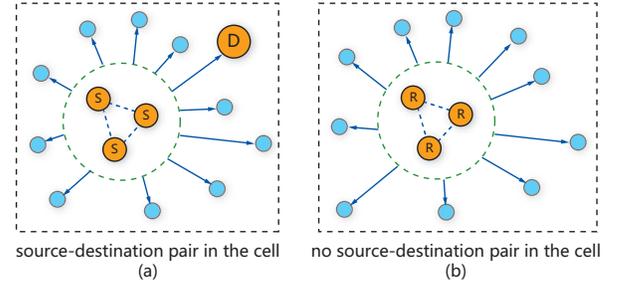


Fig. 2: Illustration of our many-to-one cooperative scheme 2. Figure (a) shows the case where there are at least one sources-destination pairs in the network. Figure (b) shows the case where there are no sources-destination pairs in the cell.

of these packets in its receiving buffer and the coefficient vectors of these k combinations are linearly independent of each other.³

According to Lemma 6, the central problem arises: *how long does it take for a destination node to receive at least $\Theta(k)$ combination on average?* If denoting the whole time as D_N , then $\mathbb{E}[D_N] \leq \mathbb{E}[D_1] + \mathbb{E}[D_2]$, where $\mathbb{E}[D_1]$ and $\mathbb{E}[D_2]$ represent the time required for all nodes in the network to have one ‘‘packet’’ of the sources belonging to that destination and the time required for the destination to receive $\Theta(k)$ packets given that all the other nodes already hold an ‘‘packet’’, respectively.

We focus on $\mathbb{E}[D_1]$ first. Since each destination has k sources, for one session, only k nodes initially hold the original packets. Letting all the other nodes in the network get these k packets is equivalent to flooding k packets to all the other nodes given that there are originally k distinct nodes holding each of these packets.

First, consider a case where k distinctive packets are stored in k nodes initially and all the other nodes in the network are empty. Now we first analyze the delay required on the process for letting all the nodes in the network have these packets.

Denote J_t as the number of nodes holding the packets from the k nodes at time t . Note that $J_0 = k$. Let $\beta_t = J_t/J_{t-1}$ represent the growth factor after one time slot. Obviously, we have

$$\beta_{t+1} = \frac{J_t + a_1 + a_2 + \dots + a_{J_t}}{J_t}, \quad (16)$$

where a_i represents the number of new nodes to which the i th packet-holding node transmits during one slot. As the number of packet-holding nodes grows, β_t yields different scale. Notice that β_t is also influenced by network division. Jointly consider these two factors, we discuss in the following two cases :

- 1) If $\frac{n}{M} = \Omega(k)$, then $k_c = \Theta(k)$. In this case, initially, J_t is much smaller than $\frac{n}{M}$ and there are on average one

³In MANETs, a training sequence is contained in each packet. A destination can then recover channel status information (CSI) through these training sequences. We assume the sequences are of the equal size of a packet and the packet size is sufficiently small compared to the total number of nodes in the network. Thus, the mobility of nodes during the acquirement of a training sequence is negligible compared to data transmission time.

such packet-holding node in each cell. As J_t grows to $\frac{n}{M}$, the number of cells where packet-holding nodes are located grows to the equal order of $\frac{n}{M}$, which guarantees that there are on average $\Theta\left(\frac{J_t M}{n}\right)$ such packet-holding nodes per cell per time slot.

- 2) If $\frac{n}{M} = O(k)$, then $k_c = \Theta\left(\frac{n}{M}\right)$: In this case, the initial number of packet holding nodes k is already much larger than the number of cells $\frac{n}{M}$. Thus, the average number of packet-holding nodes per cell is $\Theta\left(\frac{kM}{n}\right)$.

Then, we have the following lemmas:

Lemma 7: In case 1, for each destination, the time required for all nodes in the network to have one “packet” from the sources corresponding to that destination is

$$\mathbb{E}[\mathcal{D}_1] = \Theta\left(\frac{\log\left(\frac{n}{k}\right)}{M^\epsilon \log(1+M)}\right) + \Theta\left(1 + \frac{1 + \log(n - \frac{n}{M})}{M-1}\right), \quad (17)$$

where ϵ is an arbitrarily small value greater than zero.

Proof: See Appendix. ■

Similarly, for case 2, we have the following lemma:

Lemma 8: In case 2, for each destination, the time required for all nodes in the network to have one “packet” from the sources corresponding to that destination is

$$\mathbb{E}[\mathcal{D}_1] = \Theta\left(1 + \frac{n(1 + \log(n-k))}{M(n-k)}\right). \quad (18)$$

As for $\mathbb{E}\{D_2\}$, it is easy to know that it takes a single destination k slots to receive k distinctive “encoded” packets given that all the nodes in the network already hold one of them. Due to round-robin transmission, $\mathbb{E}\{D_2\} = \Theta(Mk)$. Therefore, $\mathbb{E}[D_N] \leq \mathbb{E}[D_1] + \mathbb{E}[D_2]$. Hence,

$$\mathbb{E}[D_N] = \begin{cases} \Theta(\log n) & \text{if } k = o(\log n) \\ \Theta(Mk) & \text{if } k = \omega(\log n) \end{cases}. \quad (19)$$

Remark 1: It remains an interesting future work that how decoding failure can influence the delay if $k = \Theta(n)$.

2) *Analysis on Throughput:*

Lemma 9: Under our many-to-one cooperative scheme 2, we can achieve a per-node converge-cast throughput of

$$\lambda = \begin{cases} \Theta\left(\frac{1}{\log n}\right) & \text{if } k = o(\log n) \\ \Theta\left(\frac{1}{M}\right) & \text{if } k = \omega(\log n) \end{cases}. \quad (20)$$

in a MANET.

Proof: Viewing from the perspective of a source node, it belongs to k distinctive destinations on average. Thus, it has to transmit at least k times. Then, the number of transmissions per time slot is $k/(\mathbb{E}(D_N))$. According to our delay presented in Equation (19), we obtain the per-node throughput shown in Equation (20). ■

Notice that both the throughput and delay are optimized when $M = \Theta(1)$, which renders the results presented in Theorem 2.

D. The Advantage of Our Cooperative Schemes

In static network, our many-to-one cooperative scheme allows for concurrent transmission, which converts the interfering signals into useful ones. This reduces the interference level to an extensive degree and therefore undoubtedly leads to significant improvement on throughput. When the number of layers h is sufficiently large, the aggregate throughput achieved is up to $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$, with only a $\log(n)$ factor difference compared with the upper-bound. In MANETs, with further observation on our scheme 2, we can find it is to some extent equivalent to a “flooding” algorithm but with more intelligent transmissions. However, in previous flooding algorithm, packets are simply broadcasted arbitrarily to other nodes in the cell, regardless of whether the receivers are destinations of those packets. This undoubtedly leads to some unnecessary waste on the number of transmissions, which incurs sacrifice on throughput.

E. Delay-throughput Tradeoff

In this subsection, we consider delay-throughput tradeoff obtained under our schemes.

Static network: By THEOREM 1, we obtain the delay/throughput tradeoff, as is shown as follows:

$$\begin{cases} \tilde{\Theta}\left(n^{\frac{2h^2-4h+4}{2h-1}} \cdot k^{-\frac{2h^2-2h}{2h-1}}\right), & \text{if } k = \Omega(n^{\frac{1}{2h-2}}) \\ \tilde{\Theta}\left(n^{\frac{h^2-2h+1}{2h-1}} k^{\frac{h^2-4h+2}{2h-1}}\right), & \text{if } k = O(n^{\frac{1}{2h-2}}) \end{cases}. \quad (21)$$

Note that the tradeoff for $k = O(n^{\frac{1}{2h-2}})$ is poor compared to that for $k = \Omega(n^{\frac{1}{2h-2}})$. In other words, a larger k helps to reduce delay. This is because the number of clusters in our scheme is smaller than that of sources when $k = \Omega(n^{\frac{1}{2h-2}})$. This allows more simultaneous transmitting nodes to achieve largely reduced delay but at the cost of more extra energy consumption.

MANETs: When there are M nodes in average in each cell, the delay/throughput tradeoff obtained under mobile network is M^2k . A counter-intuitive phenomenon can be observed that a smaller number of cells leads to poorer performance on both throughput and delay. However, from the perspective of MIMO, an increase on the number of nodes per cell leads to the decrease on the number of concurrent active cells, under 9-TDMA scheme. Moreover, in each cell, as the number of nodes becomes large, each node have to endure a longer waiting time before transmission. Both of the two factors reduces the efficiency, which therefore leads to a larger delay for completing the whole process. Hence, the tradeoff is optimized when $M = \Theta(1)$, with per-node throughput achieves $\Theta(1)$ and the corresponding delay reduced to $\Theta(k)$. Because it can guarantee the maximum number of concurrent active cells as well as the shortest waiting time endured by each node in the cell before transmission or reception.

F. Covering to Other Traffic Patterns

Since converge-cast can be treated as a generalized reversed “multicast”, other traffic patterns such as unicast, multicast

and broadcast can be treated as special cases of it. In this subsection, we will consider the scaling laws in such traffic patterns when applying our schemes to them.

1) *Covering to Unicast*: If we set the number of sources per session k , under converge-cast as 1, then we get unicast traffic pattern. Actually, unicast is a special type of converge-cast. Therefore, all the results obtained under converge-cast in both static and mobile networks can be easily applied to unicast in this paper.

2) *Covering to Multicast*: **Static network**: Consider step 1 and step 2 in our many-to-one cooperative scheme 1, since the process in these two steps is many-to-one and convergent, it cannot be simply reversed. Moreover, the data information from the k sources may be different, none can be treated as a duplication of those from other sources. Thus, the hierarchical scheme proposed in static network cannot be applied to multicast traffic pattern.

MANETs: Due to the mobility of nodes, spanning tree is not needed in routing establishment. Thus, our many-to-one scheme 2 can easily be applied to multicast mode, only with some minor modifications in our scheme. That is, in a cell, we figure out the maximum number of nodes that hold the packets from the same source. These nodes are picked out and then they jointly broadcast their packets in the cell. Initially, only the source holds the packet and when there are several source-destinations pairs in the same cell, we randomly pick out one such pair and let the source broadcast its packet to its destinations as well as other nodes in the cell. The algorithm continues until all the destinations receive the packet they need.

Next we turn to the analysis on throughput and delay for multicast. Since the number of packet-holding nodes grows from 1 to n rather than k to n in converge-cast, the total flooding time of a packet is bounded by $\Theta(\log n)$. Moreover, since k destinations receive the same packets, each of them can immediately decode the packet after it receives only one copy. And this process is already contained in the ‘‘flooding’’ one. Hence, we can achieve a delay of $\mathbb{E}[D_N] = \Theta(\log n)$ for multicast. As for the per-node throughput, we consider from the perspective of a source node. It should transmit one packet but has to duplicate it n times so that all the nodes in the network can get one copy. Hence, the per-node throughput yields to $\lambda = \frac{1}{\mathbb{E}[D_N]} \cdot \frac{1}{n \log n} = \frac{1}{n \log n}$.

3) *Covering to Broadcast*: Since broadcast can be treated as a special type of multicast, we know that the results obtained under static network cannot cover broadcast case. Moreover, in MANETs, with similar analysis for multicast, we obtain the same results as that listed in IV-F2.

4) *Comparison with Previous Work and Generalization*: For static network, when applying to unicast, our scheme still can achieve an aggregate throughput of $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$. This is identical to that achieved in [14] while our delay is much larger. This is because the amount of data exchange in our scheme is much larger. Hence, if concerned with delay priority, our scheme is not optimal for unicast to achieve a small delay. Next, consider the extreme case where $k = \Theta(n)$.

The aggregate throughput is still close to $\Theta(n)$ with the delay reduced to $\Theta\left(n^{\frac{4-2h}{2h-1}}\right)$. There turns out to be a significant improvement on capacity, compared with previous results in [15], [16] and [17]. In [15], the aggregate capacity scales as $\Theta(\log n)$ as n goes to infinity while in [16], the maximum rate for a collected network do not exceed $\Theta\left(\frac{1}{\log n}\right)$. In [17] where all the nodes in the network flow their data a common sink, the authors demonstrate that total data aggregation rates of $\Theta(\log n)$ and $\Theta(1)$ are optimal when operating in fading environments with power path-loss exponents that satisfy $2 < \alpha < 4$ and $\alpha > 4$, respectively. Our result also achieves a gain of $\Theta(\log n)$ compared with that in [18], where the capacity of data collection is $\Theta\left(\frac{n}{\log n}\right)$ if there are n sinks with each one collecting data from all the rest of the nodes in the network.

For MANETs, **1. Unicast**: Our extended results achieve the per-node throughput of $\Theta\left(\frac{1}{\log n}\right)$ with a $\Theta(\log n)$ delay. A gain of n is achieved on throughput compared with that obtained in [6], where the per-node throughput is $\Theta\left(\frac{1}{n}\right)$ while the delay is also $\Theta(\log n)$. The improvement on throughput is due to our intelligent cooperation between nodes with the help of MIMO. Multiple nodes can transmit simultaneously to other nodes. And a node can successfully decode the original packet once it receives only one combination. Nevertheless, redundant transmission still has to be wasted in [6] even with the adoption of network coding since a destination can decode the original packets only when it receives $\Theta(k)$ packets. **2. Multicast and broadcast**: We obtain the per-node throughput of $\Theta\left(\frac{1}{n \log n}\right)$ and the delay of $\Theta(\log n)$ under both traffic patterns. The result is close to that of [6] with only a $\log n$ factor. It is easy to understand since in such traffic patterns, a source sends identical information to several (or all) destinations. In the scheme of both [6] and ours, although all the destinations can receive the information from their common source within $\log n$ delay, a source has to endure several times’ duplication. Thus, a source has only one packet to share with all its destinations along with several data copies, which degrades the throughput performance.

The comparison verifies that all our results well cover those from the previous work and our proposed schemes are promising for not only converge-cast but other traffic patterns as well.

V. CONCLUSION

In this paper, with MIMO, we design two different cooperative schemes for static and mobile ad hoc wireless networks (MANETs), respectively. The hierarchical cooperation scheme under static networks can achieve an aggregate throughput of $\Omega(n^{1-\epsilon})$ for any $\epsilon > 0$. The scheme under MANETs features on joint multiple transmission and reception without hierarchical operations. With optimal network division in the scheme, the achievable per-node throughput can be $\Theta(1)$ with the corresponding delay reduced to $\Theta(k)$. Moreover, we find all the results derived from converge-cast in this paper under mobile networks can be easily applied to unicast, multicast and broadcast.

VI. ACKNOWLEDGMENT

This work is supported by National Fundamental research grant (2011CB302701, 2010CB731803, 2009CB3020402), NSF China (No. 60702046, 60832005, 60972050, 60632040); China Ministry of Education (No. 20070248095); China Ministry of Education Fok Ying Tung Fund (No. 122002); Qualcomm Research Grant; China International Science and Technology Cooperation Programm (No. 2008DFA11630); PUJIANG Talents (08PJ14067); Shanghai Innovation Key Project (08511500400); National Key Project of China (2009ZX03003-006-03, 2009ZX03002-003, 2009ZX03002-005, 2010ZX03003-001-01); National High tech grant of China (2009AA01Z248, 2009AA1Z249, 2009AA011802), a grant from University of Nebraska Lincoln.

REFERENCES

- [1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Inform. Theory*, Vol. 46, No. 2, pp. 388-404, Mar. 2000.
- [2] M. Grossglauser and M. D. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Networks," in *Proc. IEEE Infocom 2001*, Anchorage, Alaska, Apr. 2001.
- [3] M. Neely and E. Modiano, "Capacity and Delay Tradeoffs for Ad-Hoc Mobile Networks," *IEEE Trans. Inform. Theory*, Vol. 51, No. 6, pp. 1917-1937, June 2005.
- [4] X. Lin and N. Shroff, "The Fundamental Capacity-Delay Trade-off in Large Mobile Ad Hoc Networks," in *Third Ann. Med-Hoc-Net. Workshop*, 2004.
- [5] P. Li, Y. Fang and J. Li, "Throughput, Delay, and Mobility in Wireless Ad Hoc Networks," in *Proc. IEEE Infocom 2010*, San Diego, USA, Mar. 2010.
- [6] C. Zhang, Y. Fang, X. Zhu, "Throughput-Delay Tradeoffs in Large-Scale MANETs with Network Coding," in *Proc. IEEE Infocom 2009*, Rio de Janeiro, Brazil, Apr. 2009.
- [7] W. Huang, X. Wang, Q. Zhang, "Capacity Scaling in Mobile Wireless Ad Hoc Network with Infrastructure Support," to appear in *IEEE ICDCS 2010*, Genoa, Italy, 2010.
- [8] P. Li, C. Zhang and Y. Fang, "Capacity and delay of hybrid wireless broadband access networks," in *IEEE J. Sel. Areas Commun.*, vol. 27, No. 2, pp. 117-125, Feb. 2009.
- [9] S. Aeron and V. Saligrama, "Wireless ad hoc networks: strategies and scaling laws for the fixed snr regime", in *IEEE Trans. Inf. Theory*, vol. 53, No. 6, pp. 2044-2059, Jun. 2007.
- [10] P. Jacquet and G. Rodolakis, "Multicast Scaling Properties in Massively Dense Ad Hoc Networks," in *Proc. International Conference on Parallel and Distributed Systems*, July 2005.
- [11] X. Li, Y. Liu, S. Li, and S. Tang, "Multicast Capacity of Wireless Ad Hoc Networks Under Gaussian Channel Model", in *IEEE/ACM Trans. Networking (TON)*, Vol. 18, No. 4, pp. 1145-1157, Aug. 2010.
- [12] S. Li, Y. Liu, and X. Li, "Capacity of Large Scale Wireless Networks Under Gaussian Channel Model", in *ACM MobiCom 2008*, San Francisco, USA, Sept. 2008.
- [13] C. Hu, X. Wang, D. Nie and J. Zhao, "Multicast Scaling Laws with Hierarchical Cooperation", in *IEEE Infocom 2010*, San Diego, USA, Mar. 2010.
- [14] A. Özgür and O. Lévêque, "Throughput-Delay Trade-Off for Hierarchical Cooperation in Ad Hoc Wireless Networks", in *Proc. Int. Conf. Telecom.*, Jun. 2008.
- [15] H. E. Gamal, "On the scaling laws of dense wireless sensor networks: the data gathering channel", in *IEEE Trans. Inform. Theory*, Vol. 51, No. 3, pp. 1229-1234, Mar. 2005.
- [16] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks", in *IEEE J. Sel. Areas. Commun.*, Vol. 23, No. 4, pp. 755-764, Apr. 2005.
- [17] R. Zheng and R. J. Barton, "Toward Optimal Data Aggregation in Random Wireless Sensor Networks", in *IEEE Infocom 2007*, Anchorage, Alaska, USA, May 2007.
- [18] S. Chen, Y. Wang, X. Li and X. Shi, "Order-Optimal Data Collection in Wireless Sensor Networks: Delay and Capacity", in *IEEE SECON 2009*, June 2009.

APPENDIX

Proof of Lemma 7: For case 1, it is obvious that $\mathbb{E}\{a_i\} \approx M$. Then we have $\mathbb{E}\{\beta_{t+1}|J_t\} = \frac{J_t + J_t \mathbb{E}\{a_i\}}{J_t} \approx 1 + M$. Since $J_t = \beta_0 \beta_1 \beta_2 \dots \beta_t$, $\mathbb{E}\{J_t\} \approx k(1 + M)^t$. Note that the period ends when $J_t = \frac{n}{M}$. Denoting D_{1_1} as the time required for k packet-holding nodes to grow to $\frac{n}{M}$, then $\mathbb{P}\mathbb{r}[T_{1_1} > t|J_t] \geq \left(1 - \frac{M}{n}\right)^{tJ_t}$. Moreover, we have $\mathbb{E}\{D_{1_1}\} \geq t \mathbb{P}\mathbb{r}[D_{1_1} > t] = t \mathbb{E}_{J_t} \{\mathbb{P}\mathbb{r}[D_{1_1} > t|J_t]\} \geq t \mathbb{E}_{J_t} \left\{ \left(1 - \frac{M}{n}\right)^{tJ_t} \right\} \geq t \left(1 - \frac{M}{n}\right)^{t \mathbb{E}_{J_t}}$. The above inequality holds for all $t > 0$. We choose t to be a base $(1 + M)$ logarithm: $t \triangleq \frac{1}{M} \log_{1+M} \left(\alpha \left(\frac{n}{k}\right)^{M^\delta} \right)$, where α is chosen as $1 \leq \alpha \leq (M + 1)$ and δ is any constant number less than 1. This implies $\lim_{n \rightarrow \infty} \mathbb{E}\{D_{1_1}\} \geq \frac{\alpha \log\left(\frac{n}{k}\right)}{M^{1-\delta} \log(1+M)} \rightarrow \Theta\left(\frac{\log\left(\frac{n}{k}\right)}{M^\epsilon \log(1+M)}\right)$, $\epsilon \rightarrow 0$.

When J_t grows to $\frac{n}{M}$, let m denote the number of nodes which do not initially have the packet ($m \leq n - \frac{n}{M}$) and label these m nodes with $\{x_1, x_2, \dots, x_m\}$. Let X_i represent the number of time slots it takes for the non-packet holding node x_i to reach a cell containing a packet-holding node. The probability that at least one of the new node enters the same cell as packet-holding node x_i is $\varphi > 1 - \left(1 - \frac{M}{n}\right)^{n - \frac{n}{M}} \geq 1 - e^{1-M}$. At all times X_i are *i.i.d.* Denoting D_{1_2} as the time to expand the number of packet-holding nodes from $\frac{n}{M}$ to n , then the random variable D_{1_2} is equal to the maximum value of at most $m = \lfloor \frac{n}{M} \rfloor$ *i.i.d.* variables. Hence, $\mathbb{E}[D_{1_2}] \leq \mathbb{E}\{\max\{X_1, X_2, \dots, X_m\}\}$. Now we consider new random variables $\{Y_1, Y_2, \dots, Y_m\}$ which are assumed to be *i.i.d.* distributed with rate $\nu = \log(1/(1 - \varphi))$. Thus, $\mathbb{E}[D_{1_2}] \leq 1 + \mathbb{E}\{\max\{Y_1, Y_2, \dots, Y_m\}\}$. Denoting I_i as the interval between the $(i - 1)$ th and i th completion time. Then $\mathbb{E}\{I_1 + I_2 + \dots + I_m\} = \frac{1}{\nu} \sum_{j=1}^m \frac{1}{j}$. Hence, $\mathbb{E}[D_{1_2}] \leq 1 + \frac{1}{\nu} \sum_{j=1}^m \frac{1}{j} \leq \mathbb{E}[D_{1_2}] \leq 1 + \frac{1}{\nu} (1 + \log m) \leq 1 + \frac{(1 + \log(n - \frac{n}{M}))}{M - 1}$. And $\mathbb{E}[\mathcal{D}_1] = \mathbb{E}[D_{1_1}] + \mathbb{E}[D_{1_2}]$.