

# Converge-Cast with MIMO

Luoyi Fu<sup>1</sup>, Yi Qin<sup>2</sup>, Xinbing Wang<sup>3</sup>, Xue Liu<sup>4</sup>

<sup>1,2,3</sup>Depart. of Electronic Engineering, Shanghai Jiaotong University, China

<sup>4</sup> Depart. of Computer Science & Engineering, University of Nebraska Lincoln, USA

Email: <sup>1,2,3</sup>{yiluofu, qinyi\_33xwang8}@sjtu.edu.cn, <sup>4</sup>xueliu@cse.unl.edu

November 21, 2010

## 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.

## 1 Introduction

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  $\Theta(\frac{1}{\sqrt{n \log n}})$ \* for random network, capacity analysis of ad hoc networks have triggered

---

\*We use the following notation throughout our paper:

$$f(n) = o(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0,$$

$$f(n) = \omega(g(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \frac{g(n)}{f(n)} = 0,$$

$$f(n) = O(g(n)) \Leftrightarrow \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty,$$

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]- [6]) or transmit through coding or infrastructure support ([7]- [9]).

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.* [10] who presented a MIMO collaborative strategy which achieves a per-node capacity of  $\Theta(n^{-1/3})$ . Following that, Özgür *et al.* [16] 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 [15] 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 [11]- [14].

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 [17], [18], [19], [20]), 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 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

---


$$f(n) = \Omega(g(n)) \Leftrightarrow \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty,$$

$$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n)) \text{ and } g(n) = O(f(n))$$

$f(n) = \tilde{\Theta}(\cdot)$ : The corresponding order  $\Theta(\cdot)$  which contains a logarithmic order.

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 [7] in MANET and [16], [17]- [20] 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 and outline the directions for future work in Section 7. Some proofs are provided in-line and others are in Appendix.

## 2 Models and Definitions

### 2.1 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. We do not consider the case where each node has multiple antennas for the following two reasons: 1. If each node is assumed to have constant bounded number of antennas, say,  $c$  antennas, then the throughput is  $c$  times that achieved in single-antenna case, which does not change the throughput order; 2. If each node has  $n_r$  antennas where  $n_r$  scales with  $n$ , then the throughput achieved in order sense is  $n_r$  times that of single-antenna case. This is trivial and assuming  $n_r$  antennas on one node is not realistic.

### 2.2 Definitions

*Converge-cast Session:* a converge-cast session is defined as the set composed of one destination and its corresponding  $k$  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  $\lambda$  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  $\Lambda = n\lambda$ .

### 2.3 Notations

In table 1, we list all the parameters that will be used in later analysis, proofs and discussions.

Table 1: 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.

## 3 Many-to-one Scheme under Static Networks

In this section, we will design a cooperative scheme with MIMO under static networks. Then we will analyze the throughput and delay achieved under the scheme.

### 3.1 Many-to-one Cooperative Scheme 1 under Static Networks

As is shown in [16], hierarchical cooperation can achieve better throughput scaling than classical multihop schemes under certain assumptions on the channel model in static wireless network. This

motivates us to design a hierarchical scheme which can be applied to converge-cast.

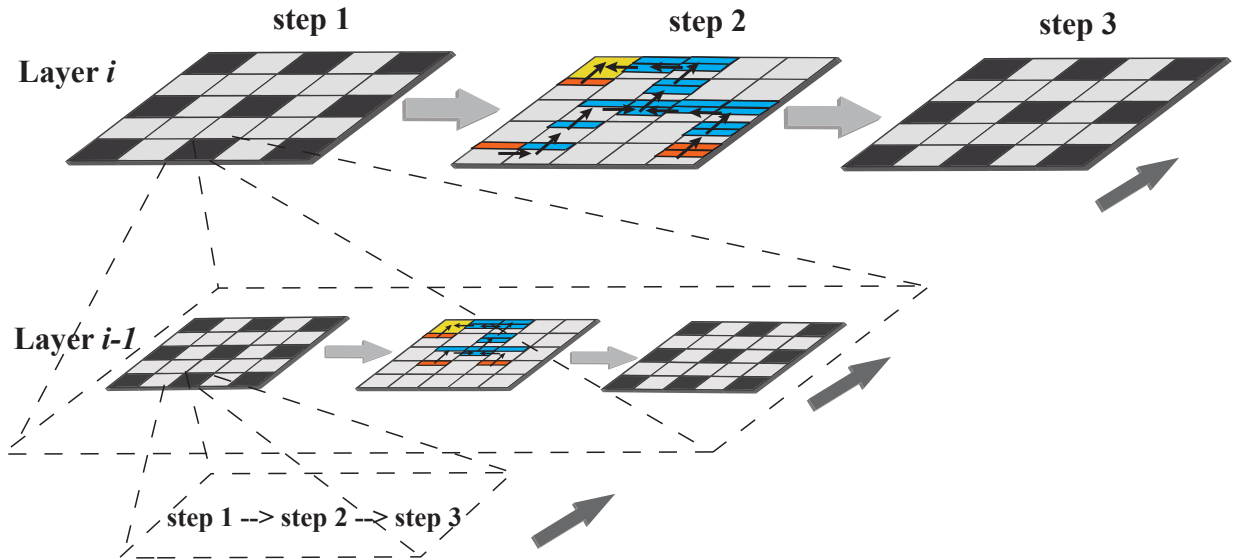


Figure 1: A global view of our hierarchical many-to-one cooperative scheme. The algorithm starts from the bottom layer and keeps executing until it reaches layer  $h$ .

### 3.1.1 Scheduling Algorithm

Under hierarchical schemes, a network is divided into clusters with equal number of nodes in each one. 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 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  $i$ . At each layer, the scheme is divided into three steps, which are described as follows. And Figure 1 shows the hierarchy and recursion between layers in our algorithm.

**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 packets<sup>†</sup> to other nodes in the same cluster. For each node in the cluster, the  $k_{i-1}$  sources jointly transmit their packets to other nodes in the cluster, which receives a linear combination of that bit mixed with channel coefficients. The process keeps until all the other 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

<sup>†</sup>Note that the size of packet in our cooperative scheme 1 differs at different layers. Each node divide a packet into  $\frac{n_{i-1}}{k_i}$  packets. Each of these packets is then used in the next layer  $i - 1$ .

node, the process is a 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. Note that our algorithm starts from the bottom layer, *i.e.*, layer 1 of the network and continues to a higher layer until layer  $h$ .

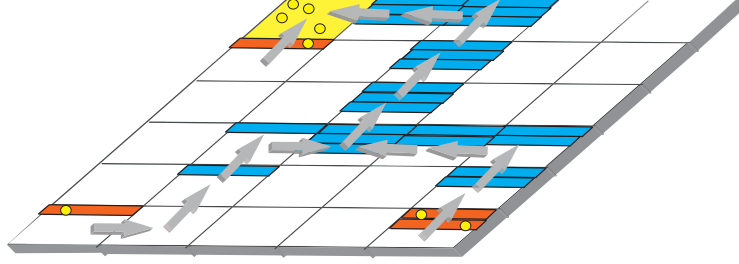


Figure 2: An example of a CT in multi-hop MIMO transmission. Assume one destination has 4 sources. The red parts represent source clusters. 1/4 percent of the nodes are allowed for transmission in the two source clusters on the left while 1/2 percent of the nodes are active in the source cluster on the 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.

**Step 2. Multi-hop MIMO Transmissions:** For the sake of energy efficiency, we construct a multi-hop routing mode rather than adopting direct MIMO transmission between clusters in this step. 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 unit 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,  $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{\overline{uw}\}$ ,  $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{\overline{vw}\}$ ,  $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} - \{\overline{uv}\}$ .
3. Obtaining the MT( $P_i$ ) 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  $\mathcal{E}_{CT} \rightarrow \mathcal{E}_{CT} \cup E(u, w)$ ,  $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{\overline{uv}\}$ . Finally,  $\mathcal{E}_{CT}$  is the set of edges of CT( $P_i$ ).

Figure 2 shows a simple example of the data flow on such converge-cast tree (CT).

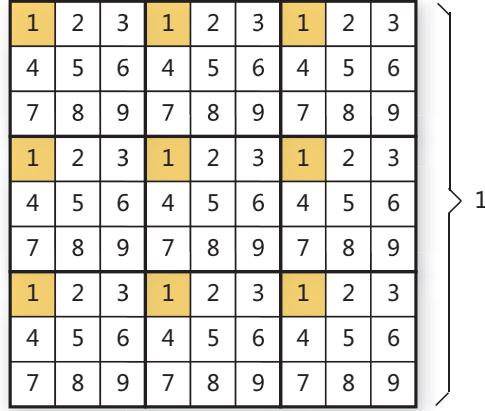


Figure 3: 9-TDMA scheme where the whole network is divided into clusters with equal area. Each 9 groups are categorized as a group. All the yellow cells in each group (numbered with 1) can transmit simultaneously in a time slot. In the next time slot all the cells numbered with 2 transmit and so on.

**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, as is shown in Figure 3.

### 3.1.2 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 the upper-bound of throughput and our main results as follows.

**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*

$$\sum_{i=1}^n \lambda_i \leq Cn \log n \quad (3)$$

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

**Proof 1** *Provided in Appendix A.*

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

$$\Lambda = \widetilde{\Theta} \left( n^{\frac{2h-2}{2h-1}} \cdot k^{-\frac{1}{2h-1}} \right) \quad (4)$$



with the delay of

$$\mathbb{E}[T] = \begin{cases} \tilde{\Theta} \left( n^{\frac{2h^2-4h+3}{2h-1}} \cdot k^{-\frac{2h^2-2h-1}{2h-1}} \right), & \text{if } k = \Omega(n^{\frac{1}{2h-2}}) \\ \tilde{\Theta} \left( n^{\frac{h^2-2h+2}{2h-1}} k^{\frac{h^2-4h+3}{2h-1}} \right), & \text{if } k = O(n^{\frac{1}{2h-2}}) \end{cases} \quad (5)$$

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

**Lemma 2** (Lemma 1 in [15]) *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.<sup>‡</sup>, for any  $\delta > 0$ .*

**Lemma 3** (Lemma 4.3 in [15]) *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 4** *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 2** *The proof is similar to that of Lemma 4.5 in [15] and we do not present the proof here.*

**Lemma 5** *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 3** *The proof is similar to that of Lemma 4.6 in [15] 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  $\tilde{\Theta}(n_{i-1}^a k_{i-1}^b)$  is achievable at layer  $i-1$  w.h.p., where  $0 \leq a \leq 1$ ,  $-1 \leq b \leq 0$  and  $a+b < 0$ . It is easy to obtain that the total time to complete  $k_i t_i$  traffic loads is

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

---

<sup>‡</sup>We say an event occurs with high probability (w.h.p.) if its probability goes to 1 as  $n \rightarrow \infty$ .

Hence, the throughput can be expressed as

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

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

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

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  $T_i = k_i^{\frac{-b}{a+b-2}} n_i^{\frac{b-1}{a+b-2}}$ , the recursion calculation is listed as follows:

$i$	$a$	$b$	$n_i$	$T_i$
2	0	-1	$k_2^{\frac{1}{2}} n_1^{\frac{3}{2}}$	$k_2^{-\frac{1}{3}} n_2^{\frac{1}{2}}$
3	$\frac{2}{3}$	$-\frac{1}{3}$	$k_3^{\frac{1}{3}} n_2^{\frac{5}{4}}$	$k_3^{-\frac{1}{5}} n_3^{\frac{4}{5}}$
			⋮	
			⋮	
			⋮	
$h$	$\frac{2h-4}{2h-3}$	$-\frac{1}{2h-3}$	$k_h^{\frac{1}{2h-2}} n_{h-1}^{\frac{2h-1}{2h-2}}$	$k_h^{-\frac{1}{2h-1}} n_h^{\frac{2h-2}{2h-1}}$

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

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

In the optimized result,  $k = \Omega\left(n^{\frac{a-1}{a-2}}\right) = \Omega\left(n^{\frac{1}{2h-2}}\right)$ .

Moreover, since  $n_i = k_i^{\frac{1}{2i-2}} n_{i-1}^{\frac{2i-1}{2i-2}}$  and  $k_{i-1} = \tilde{\Theta}\left(\frac{k_i}{n_{c_i}}\right) = \tilde{\Theta}\left(\frac{k_i}{n_i} \cdot n_{i-1}\right)$ , we have

$$\frac{k_i}{n_i} = \tilde{\Theta}\left(\frac{k_{i-1}}{n_{i-1}}\right)$$

and

$$\left(\frac{n_i}{k_i}\right) \cdot n_i^{2i-3} = n_{i-1}^{2i-1}.$$

And this yields to the following derivation:

$$\begin{aligned} n_{i-1} &= \left(\frac{n_i}{k_i}\right)^{\frac{1}{2i-1}} \cdot n_i^{\frac{2i-3}{2i-1}} \\ &= \left(\frac{n}{k}\right)^{\frac{1}{2i-1}} \cdot n_i^{\frac{2i-3}{2i-1}} \\ &= \left(\frac{n}{k}\right)^{\frac{1}{2i-1}} \cdot \left(\left(\frac{n}{k}\right)^{\frac{1}{2i+1}} \cdot n_{i+1}^{\frac{2i-1}{2i+1}}\right)^{\frac{2i-3}{2i-1}} \\ &= \left(\frac{n}{k}\right)^{\frac{1}{2i-1}} \cdot \left(\frac{n}{k}\right)^{\frac{1}{2i+1} \frac{2i-3}{2i-1}} \cdot n_{i+1}^{\frac{2i-3}{2i+1}} \\ &= \left(\frac{n}{k}\right)^{\frac{1}{2i-1} + \frac{1}{2i+1} \frac{2i-3}{2i-1} + \frac{1}{2i+3} \frac{2i-3}{2i+1} \dots} \cdot n^{\frac{2i-3}{2h-1}}. \end{aligned}$$

Hence,

$$\begin{aligned} n_1 &= \left(\frac{n}{k}\right)^{\frac{1}{3} + \left(\frac{1}{5} \frac{1}{3} + \frac{1}{7} \frac{1}{5} \dots \frac{1}{2h-1} \frac{1}{2h-3}\right)} n^{\frac{1}{2h-1}} \\ &= \left(\frac{n}{k}\right)^{\frac{1}{3} + \frac{1}{2} \left(\frac{1}{3} - \frac{1}{5} + \frac{1}{5} - \frac{1}{7} \dots + \frac{1}{2h-3} - \frac{1}{2h-1}\right)} n^{\frac{1}{2h-1}} \\ &= \left(\frac{n}{k}\right)^{\frac{1}{3} + \frac{1}{2} \left(\frac{1}{3} - \frac{1}{2h-1}\right)} n^{\frac{1}{2h-1}}, \end{aligned}$$

from which we can obtain  $n_1 = n^{\frac{h}{2h-1}} k^{-\frac{h-1}{2h-1}}$ .

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

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

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  $k_i = O \left( n_i^{\frac{a-1}{a-2}} \right)$ .

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. Dividing the network in the optimal way at each layer, the relationship between  $n_i$ ,  $k_i$  and throughput  $T_i$  is  $n_i = k_i^{-1} n_{i-1}^{3-2a}$   $T_i = k_i^{\frac{a-1}{3-2a}} n_i^{\frac{2-a}{3-2a}}$ . The recursive calculation is listed as follows:

$i$	$a$	$b$	$n_i$	$T_i$
2	0	0	$n_1^3$	$n_2^{\frac{2}{3}} k_2^{-\frac{1}{3}}$
3	$\frac{2}{3}$	$-\frac{1}{3}$	$n_2^{\frac{5}{3}}$	$n_3^{\frac{4}{5}} k_3^{-\frac{1}{5}}$
			⋮	
			⋮	
			⋮	
$h$	$\frac{2h-4}{2h-3}$	$-\frac{1}{2h-3}$	$k_h^{-1} n_{h-1}^{\frac{2h-1}{2h-3}}$	$k_h^{-\frac{1}{2h-1}} n_h^{\frac{2h-2}{2h-1}}$

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.

## 4 Many-to-one Scheme under MANETs

In Section 3, 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.

### 4.1 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.

### 4.2 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.

#### 4.2.1 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 3 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 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 4.

### 4.3 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:

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

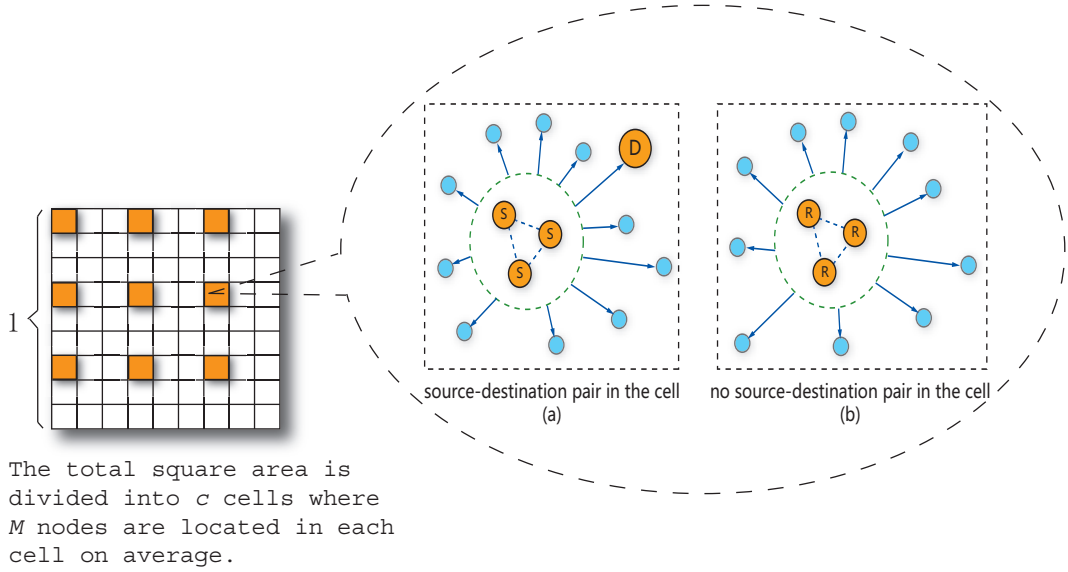


Figure 4: Illustration of our many-to-one cooperative scheme 2. The unit square is divided into  $c$  cells with  $M$  nodes located in each cell averagely. The cells colored with yellow can be active concurrently under 9-TDMA scheme. For an active cell, what may be going on is shown in the subfigures (a) and (b) on the right. (a) shows the case where there are at least one source-destination pairs in the network. (b) shows the case where there are no sources-destination pairs in the cell.

To prove Theorem 2, we turn to the proof for delay in 4.3.1 first and then prove the throughput in 4.3.2.

#### 4.3.1 Analysis on Delay

Before the proof of delay, we first introduce the following two lemmas.

**Lemma 6** 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  $\lim_{n \rightarrow \infty} \frac{n}{c \log c} = \infty$ .

**Proof 4** Provided in Appendix B.

**Lemma 7** 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 of these packets in

its receiving buffer and the coefficient vectors of these  $k$  combinations are linearly independent of each other<sup>§</sup>.

Viewing from the perspective of network coding, the central problem arises: *how long does it take for a destination node to receive at least  $\Theta(k)$  combination<sup>¶</sup> 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 belong 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. Note that for each destination, it has  $k$  sources. That is to say, for one session, initially, there are only  $k$  nodes which hold the original packets. And the process for letting all the other nodes in the network get these  $k$  packets is equivalent to the process for 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  $J_0 = k$ . And 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. Note that in each time cell, it is a *multiple-transmission-multiple-reception* process. As the number of packet-holding nodes grows, the growth factor  $\beta_t$  will yield different scale. Also notice that  $\beta_t$  is also influenced by different 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  $\frac{n}{M}$ . Thus, there are on average one such packet-holding node in each cell. This node then transmits the packet to all the other  $M - 1$  nodes during one time slot. As  $J_t$  grows to  $\frac{n}{M}$ , the number of cells where packet-holding nodes are located becomes the equal order of  $\frac{n}{M}$ , which can guarantee 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  $\Theta\left(\frac{kM}{n}\right)$ .

---

<sup>§</sup>In mobile networks, to obtain the channel status information (CSI) of other nodes, a training sequence is contained in each packet. A destination can then recover CSI through these training sequence contained in each packets. We assume it is 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 can be neglected, if compared to data transmission time.

<sup>¶</sup>For a node who does not have the packet initially, it only needs to receive one combination of some (or all) of these  $k$  original packets. And for simplicity, in the following part, when we say a packet, we refer to the combination actually.

Then, we have the following lemmas:

**Lemma 8** *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*

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

where  $\epsilon$  is an arbitrarily small value greater than zero.

**Proof 5** *Provided in Appendix C.*

**Lemma 9** *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*

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

**Proof 6** *For case 2, the procedure directly starts from the stage during which the number of nodes grows from  $k$  to  $n$ . Following the same analysis as that in case 1, we obtain*

$$\mathbb{E}[\mathcal{D}_1] = \Theta\left(1 + \frac{n(1 + \log(n-k))}{M(n-k)}\right). \quad (19)$$

Now, we turn back to the real case where all the nodes in the network are destinations, each of which is required to receive packets from its corresponding  $k$  sources. Due to the fairness achieved in our scheme, each node in a cell has equal chances to receive packets. Since there are on average  $M$  nodes per cell, every node can be active once every  $M$  time slots. Therefore, the average delay of the whole network during this period is  $\mathbb{E}[D_1] = M \cdot \mathbb{E}\{\mathcal{D}_1\}$ . Now we give the following lemma related to  $\mathbb{E}[D_1]$ .

**Lemma 10** *The average delay for letting all nodes in the network to have one “packet” of the sources belonging to the same destination is bounded by*

$$\mathbb{E}[D_1] = \begin{cases} \Theta(M) & \text{if } M = \omega(\log(n)) \\ \Theta(1) & \text{if } M = o(\log(n)) \end{cases}. \quad (20)$$

**Proof 7** *For case 1 presented in Lemma 8, we can get*

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

since  $n/M = \Omega(k)$ , we have  $n/k = \Omega(M)$ . Thus, the term  $\Theta\left(\frac{\log\left(\frac{n}{k}\right)}{\log(1+M)}\right)$  is less than 1 and is negligible compared to the term  $\Theta\left(M + 1 + \log\left(n - \frac{n}{M}\right)\right)$ . Hence, we discuss on the term  $\Theta\left(M + 1 + \log\left(n - \frac{n}{M}\right)\right)$ .

- When  $M = \Theta(1)$ ,  $\log\left(n - \frac{n}{M}\right)$  approaches to zero. Thus,  $M \cdot \mathbb{E}[\mathcal{D}_1] = \Theta(M + 1) = \Theta(1)$ .
- When  $M = \omega(1)$ , the term  $\left(\log\left(n - \frac{n}{M}\right) + 1\right)$  can be omitted. Hence,  $M \cdot \mathbb{E}[\mathcal{D}_1] = \Theta(M)$ .



For case 2 presented in Lemma 9, we have

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

Since  $k = o(n)$ ,  $\left( \frac{n(1 + \log(n - k))}{n - k} \right) \approx \log(n)$ . Therefore,  $\mathbb{E}[D_1] = \Theta(M + \log(n))$ . With further discussion on  $M$  and  $\log(n)$ , we get

$$M \cdot \mathbb{E}[\mathcal{D}_1] = \begin{cases} \Theta(M) & \text{if } M = \omega(\log(n)) \\ \Theta(\log(n)) & \text{if } M = o(\log(n)) \end{cases}.$$

Jointly consider case 1 and case 2, we have

$$\mathbb{E}[D_1] = \begin{cases} \Theta(M) & \text{if } M = \omega(\log(n)) \\ \Theta(\log(n)) & \text{if } M = o(\log(n)) \end{cases}. \quad (21)$$

This completes our proof of Lemma 10.

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. And consider the fact that each destination in one cell will have such chance once every  $M$  time slots, we obtain the average delay required for all the destinations to receive  $k$  distinctive “encoded” packets is  $Mk$ . That is to say,  $\mathbb{E}\{D_2\} = \Theta(Mk)$ . Therefore, we can bound the total delay achieved under our scheme as

$$\mathbb{E}[D_N] \leq \mathbb{E}[D_1] + \mathbb{E}[D_2] = \begin{cases} \Theta(Mk) & \text{if } k = \omega(\log(n)) \\ \Theta(\log(n)) & \text{if } k = o(\log(n)) \end{cases}. \quad (22)$$

**Remark 1** During the analysis of  $\mathbb{E}\{D_2\}$ , we do not take into account the case of decoding failure. That is, a destination still cannot successfully decode the original  $k$  packets after it receives  $k$  distinctive “encoded” packets. Such failure may be caused by the fact that the rank of the coefficient matrix generated from these  $k$  “encoded” packets is less than  $k$ . In this case, the destination has to receive more extra packets in order to successfully decode the  $k$  packets. However, based on the assumption that  $k = o(n)$ , the probability of such decoding failure can be upper-bounded by a constant factor, which does not change the order of our bound on  $\mathbb{E}\{D_2\}$ . If we remove the assumption  $k = o(n)$ , then we cannot simply treat decoding failure at the destination as a constant bound and the average time spent for a single destination to successfully decode all the  $k$  packets must be larger than  $\Theta(k)$ . It remains an interesting future work that how decoding failure can influence the delay if  $k = \Theta(n)$ .

### 4.3.2 Analysis on Throughput

**Lemma 11** Under our many-to-one cooperative scheme 2, we can achieve a per-node converge-cast throughput of

$$\lambda = \begin{cases} \Theta \left( \frac{1}{\log(n)} \right) & \text{if } k = o(\log(n)) \\ \Theta \left( \frac{1}{M} \right) & \text{if } k = \omega(\log(n)) \end{cases}. \quad (23)$$

in an MANET.

**Proof 8** We calculate the throughput through the following way: 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  $k/(\mathbb{E}(D_N))$ . As can be seen from the results on delay, the term  $Mk$  dominates the scale of the total delay. And we can simply regard delay as  $\Theta(Mk)$ . Thus, we obtain the per-node throughput shown in Equation (11). This completes our proof.

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

## 5 discussion

In the previous sections, we have derived all the performance metrics under both static and mobile networks. In this section, we will further discuss these results.

### 5.1 The advantage of Our Cooperative Schemes

In this section we focus on the effect of our proposed schemes brought to the network. 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 an improvement on throughput. When the number of layers  $h$  are sufficiently large in our scheme 1, the aggregate throughput can reach  $\Theta(n)$ . This is close to the upper-bound with difference of only a  $\log(n)$  factor. 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 transmission. 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 transmission, which incurs sacrifice on throughput.

### 5.2 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:

$$\begin{cases} \tilde{\Theta} \left( n^{\frac{2h^2-4h+4}{2h-1}} \cdot k^{-\frac{2h^2-2h}{2h-1}} \right), & \text{if } k = \Omega(n^{\frac{1}{2h-2}}) \\ \tilde{\Theta} \left( n^{\frac{h^2-2h+1}{2h-1}} k^{\frac{h^2-4h+2}{2h-1}} \right), & \text{if } k = O(n^{\frac{1}{2h-2}}) \end{cases}. \quad (24)$$

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:** 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 MIMO perspective, 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.

### 5.3 Covering to Other Traffic Patterns

For MANETs, we choose the optimal network division, i.e., substituting  $M = \Theta(1)$  into both our results on converge-cast and those applied to unicast, multicast and broadcast.

#### 5.3.1 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. To present the extended results more clearly, we discuss in the following two cases:

1. Static network: in this case, when we substitute  $k = 1$  into Equation (1), we obtain the aggregate throughput for unicast, as is shown in the following equation:

$$\Lambda = \tilde{\Theta} \left( n^{\frac{2h-2}{2h-1}} \right).$$

And the delay achieved is

$$\mathbb{E}[D] = \tilde{\Theta} \left( n^{\frac{h^2-4h+3}{2h-1}} \right).$$

2. MANETs: in this case, substituting  $k = 1$  into our results, we get the delay for unicast,  $\mathbb{E}[D_N] = \Theta(\log(n))$ .

#### 5.3.2 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 the scheme . That is, in a cell, we figure out the maximum number of nodes that hold the packets from the same source. These nodes then jointly broadcast these 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 broadcasts its packet to its destinations as well as other nodes in the cell. The algorithm continues until all the destinations receives the packet they need.

Next we turn to the analysis on throughput and delay for multicast under our modified scheme 2. Consider the case where there are only one node in the network holding the packet at the beginning. Thus, the growth of the number of packet-holding nodes is from 1 to  $n$  rather than  $k$  to  $n$  under converge-cast and the total flooding time for a packet is bounded by  $\Theta(\log(n))$ . Moreover, since the packet for each of the  $k$  destinations is identical, no extra time is needed for each destination to wait for the additional  $k - 1$  packets. The destination 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

$$\mathbb{E}[D_N] = \Theta(\log(n)) \quad (25)$$

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 capacity yields to

$$\lambda = \frac{1}{\mathbb{E}[D_N]} \cdot \frac{1}{n} = \frac{1}{n}. \quad (26)$$

### 5.3.3 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 mobile network, to make the result applicable to broadcast, we only need to refer to Equation (26) and Equation (25) and setting  $k$  as  $\Theta(n)$ . Since a source has one packet to transmit with  $n$  copies, the results are the same as that of multicast listed in 5.2.2.

### 5.3.4 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 [16] 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 [17], [18] and [19]. In [17], the aggregate capacity scales as  $\Theta(\log n)$  as  $n$  goes to infinity while in [18], the maximum rate for a collected network do not exceed  $\Theta\left(\frac{1}{\log n}\right)$ . In [19] 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 [20], where the capacity of data collection is

Network	Traffic	Throughput	Delay
Static Network	Unicast	Ours: $\tilde{\Theta}\left(n^{\frac{2h-2}{2h-1}}\right)$	$\tilde{\Theta}\left(n^{\frac{h^2-4h+3}{2h-1}}\right)$
		[16]:	-
	Converge-cast	Ours: $\Theta(n)$	$\Theta\left(n^{\frac{4-2h}{2h-1}}\right)$
		[17]: $\Theta(\log n)$	-
		[18]: $\Theta\left(\frac{1}{\log n}\right)$	-
		[19]: $\Theta(\log n)$ and $\Theta(1)$	-
		[20]: $\Theta\left(\frac{n}{\log n}\right)$	-
Mobile Network	Unicast	Ours: $\Theta\left(\frac{1}{\log(n)}\right)$	$\Theta(\log(n))$
		[7]: $\Theta\left(\frac{1}{n}\right)$	$\Theta(\log n)$
	Multicast	$\Theta\left(\frac{1}{n \log(n)}\right)$	$\Theta(\log(n))$
		[7]: $\Theta\left(\frac{1}{n}\right)$	$\Theta(\log n)$
	Broadcast	$\Theta\left(\frac{1}{n \log(n)}\right)$	$\Theta(\log(n))$
		[7]: $\Theta\left(\frac{1}{n}\right)$	$\Theta(\log(n))$

Table 2: Throughput and Delay of converge-cast and that extended to unicast, multicast and broadcast under our schemes in both static and mobile networks. Comparison is also made between our results and previous ones.

$\Theta\left(\frac{n}{\log n}\right)$  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 [7], 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 [7] 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  $\Theta\left(\frac{1}{n}\right)$  and the delay of  $\Theta(\log n)$  under both traffic patterns. The result is the same as that of [7]. 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 [7] 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.

Through the comparison, it can be seen that all our results well cover and unify those from the previous work, which also demonstrates the robustness of our converged-based analysis.

## 6 Conclusion

Through the comparison, it can be seen that all our results well cover and unify those from the previous work, which also demonstrates the robustness of our converged-based analysis. 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.

There are still many aspects for us to investigate in the future. For example, it remains an interesting problem for designing a scheme with MIMO under other mobility models such as random walk and random way point models. The network performance may be quite different from that obtained under *i.i.d.* mobility model. Furthermore, it also remains to be a future work on appropriate modification of our scheme under MANET in order to make it also applicable to the scenario where all the nodes transmit their data to a single sink in the network, since our current scheme cannot guarantee a small decoding failure probability for this extreme case.

## Appendix

**A. Proof of Lemma 2:** For each destination node in the network, there are  $k$  randomly chosen sources belonging to it. If the sets of source nodes for each destination do not intersect with each other,  $nk$  nodes will serve as sources in total. However, there are only  $n$  nodes in the whole network. Thus, by treating the source-destination pair from a reverse view, for each node  $s$ , there are on average  $k$  nodes (denoting as  $d_1, d_2, \dots, d_k$ ) which choose  $s$  as one of its source nodes. Assume each source node transmits data to  $d$  at data rate  $\lambda_i$ . Since  $d$  has to receive  $k$  distinctive information from these  $k$  sources, it acts as  $k$  different nodes during each reception. Thus, the total data rate to the destination  $d$  is upper-bounded by the capacity of a multiple-input multiple-output channel between  $d$  and the rest of the network. That is,

$$\sum_{i=1}^k \lambda_i \leq k \log \left( 1 + \frac{P}{N_0} \sum_{i=1, s_i \neq d}^n \left| \frac{G}{d_{s_i d}^\alpha} \right|^2 \right). \quad (27)$$

According to Lemma 2, the distance between  $s_i$  and  $d$  is larger than  $\frac{1}{n^{1+\delta}}$  *w.h.p.* Thus,

$$\sum_{i=1}^k \lambda_i \leq k \log \left( 1 + \frac{PG}{N_0} n^{\alpha(1+\delta)+1} \right) \leq Ck \log n, \quad (28)$$

where  $C$  is a constant that does not depend on  $n$  and  $k$ .

If we assume that  $\lambda_i$ ,  $1 \leq i \leq k$  are identical, then we have

$$\lambda_i \leq C \log n, \quad (29)$$

which implies that

$$\sum_{i=1}^n \lambda_i \leq Cn \log n. \quad (30)$$

This completes our proof.

**B. Proof of Lemma 7:** Consider a particular cell. Let  $X_i$  denote the 0-1 random variable with  $X_i = 1$  representing  $v_i$  in the cell and  $X_i = 0$  representing  $v_i$  not in the cell. Note that  $X_i$  and  $X_j$  are independent with each other. Obviously,  $\mathbb{P}\mathbb{r}[X_i = 1] = 1/c$ . According to Chernoff bounds, we have

$$\mathbb{P}\mathbb{r} \left\{ \sum_{i=1}^n X_i > (1 + \delta) \frac{n}{c} \right\} \leq e^{-\delta^2 \frac{n}{3c}} \quad (31)$$

and

$$\mathbb{P}\mathbb{r} \left\{ \sum_{i=1}^n X_i < (1 - \delta) \frac{n}{c} \right\} \leq e^{-\delta^2 \frac{n}{2c}}. \quad (32)$$

Hence, with the assumption that  $\lim_{n \rightarrow \infty} \frac{n}{c \log c} = \infty$ , we have

$$\mathbb{P}\mathbb{r} \left\{ \left| \sum_{i=1}^n X_i - \frac{n}{c} \right| < \delta \frac{n}{c} \right\} \leq e^{-\theta \frac{n}{c}} < e^{-\theta \frac{n}{c \log c}} \rightarrow 0 \quad (33)$$

as  $n \rightarrow \infty$ . Thus,  $M = \frac{n}{c}$  w.h.p. This completes our proof.

**C. Proof of Lemma 8:** For case 1, it is obvious that  $\mathbb{E}\{a_i\} \approx M$ . Then we have

$$\mathbb{E}\{\beta_{t+1} | J_t\} = \frac{J_t + J_t \mathbb{E}\{a_i\}}{J_t} \approx 1 + M. \quad (34)$$

Since  $J_t = \beta_0 \beta_1 \beta_2 \dots \beta_t$ ,

$$\mathbb{E}\{J_t\} \approx k(1 + M)^t. \quad (35)$$

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

$$\mathbb{P}\mathbb{r}[T_{1_1} > t | J_t] \geq \left(1 - \frac{M}{n}\right)^{tJ_t}. \quad (36)$$

Moreover, we have

$$\begin{aligned} \mathbb{E}\{D_{1_1}\} &\geq t \mathbb{P}\mathbb{r}[D_{1_1} > t] \\ &= t \mathbb{E}_{J_t} \{ \mathbb{P}\mathbb{r}[D_{1_1} > t | J_t] \} \\ &\geq t \mathbb{E}_{J_t} \left\{ \left(1 - \frac{M}{n}\right)^{tJ_t} \right\} \\ &\geq t \left(1 - \frac{M}{n}\right)^{t \mathbb{E}_{J_t}}. \end{aligned} \quad (37)$$

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  $\alpha$  is chosen as  $1 \leq \alpha \leq (M + 1)$  and  $\delta$  is any constant number less than

1. Using this value of  $t$  in Inequality (37), we get

$$\begin{aligned}\mathbb{E}\{D_{1_1}\} &\geq \frac{\log(\alpha) + M^\delta \log(\frac{n}{k})}{M \log(1+M)} \left[ \left(1 - \frac{M}{n}\right)^n \right]^{\frac{k \left(\alpha \left(\frac{n}{k}\right)^{M^\delta}\right)^{\frac{1}{M}} \log(\frac{n}{k})}{nM \log(1+M)}} \\ &= \frac{\log(\alpha) + M^\delta \log(\frac{n}{k})}{M \log(1+M)} \left\{ \left[ \left(1 - \frac{M}{n}\right)^n \right]^{\frac{1}{M}} \right\}^{\frac{k \left(\alpha \left(\frac{n}{k}\right)^{M^\delta}\right)^{\frac{1}{M}} \log(\frac{n}{k})}{n \log(1+M)}}.\end{aligned}\quad (38)$$

Note that  $\left[\left(1 - \frac{M}{n}\right)^n\right]^{\frac{1}{M}} \rightarrow e^{-1}$  when  $n$  goes to infinity. Now we consider the term  $\frac{k \left(\alpha \left(\frac{n}{k}\right)^{M^\delta}\right)^{\frac{1}{M}} \log(\frac{n}{k})}{n \log(1+M)}$ . Obviously,

$$\frac{k \left(\alpha \left(\frac{n}{k}\right)^{M^\delta}\right)^{\frac{1}{M}} \log(\frac{n}{k})}{n \log(1+M)} < \frac{(n/k)^{M^{-(1-\delta)}} \log(n/k)}{n/k} \rightarrow 0$$

as  $n \rightarrow \infty$ . Hence, it follows that

$$\left[ \left(1 - \frac{M}{n}\right)^n \right]^{\frac{k \left(\alpha \left(\frac{n}{k}\right)^{M^\delta}\right)^{\frac{1}{M}} \log(\frac{n}{k})}{nM \log(1+M)}} \rightarrow 1.$$

This implies

$$\lim_{n \rightarrow \infty} \mathbb{E}\{D_{1_1}\} \geq \frac{\alpha \log\left(\frac{n}{k}\right)}{M^{1-\delta} \log(1+M)}. \quad (39)$$

To obtain a tight bound on Inequality (39), we choose  $\delta$  to be a value very close to 1. Thus, it can be inferred that

$$\mathbb{E}\{D_{1_1}\} = \Theta\left(\frac{\log\left(\frac{n}{k}\right)}{M^\epsilon \log(1+M)}\right), \quad (40)$$

where  $\epsilon$  is an arbitrarily small value greater than zero.

When  $J_t$  grows to  $\frac{n}{M}$ , unlike the previous process, where  $\mathbb{E}[a_i]$  almost remains unchanged per time slot, the duplicate rate  $\mathbb{E}[a_i]$  starts to vary in different time slots. 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. Due to the multi-reception,  $x_i$  must receive a packet at this time. 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 independent and identically distributed. 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  $\nu = \log(1/(1-\varphi))$ . Note that  $1 + Y_i$  is stochastically greater than  $X_i$ . 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. It is easy to know that  $I_i$  is the first completion time of  $M - i + 1$  racing exponentials. It follows that

$$\mathbb{E}\{I_1 + I_2 + \dots + I_m\} = \frac{1}{\nu} \sum_{j=1}^m \frac{1}{j}. \quad (41)$$

Hence,  $\mathbb{E}[D_{1_2}] \leq 1 + \frac{1}{\nu} \sum_{j=1}^m \frac{1}{j}$ , which is upper bounded by  $1 + \frac{1}{\nu}(1 + \log m)$ . Therefore,

$$\begin{aligned} \mathbb{E}[D_{1_2}] &\leq 1 + \frac{1}{\nu}(1 + \log m) \\ &\leq 1 + \frac{(1 + \log(n - \frac{n}{M}))}{M - 1}. \end{aligned} \quad (42)$$

And

$$\begin{aligned} \mathbb{E}[\mathcal{D}_1] &= \mathbb{E}[D_{1_1}] + \mathbb{E}[D_{1_2}] \\ &= \Theta\left(\frac{\log\left(\frac{n}{k}\right)}{M^\epsilon \log(1 + M)}\right) + \Theta\left(1 + \frac{1 + \log(n - \frac{n}{M})}{M - 1}\right). \end{aligned} \quad (43)$$

## 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 Tradeoff in Large Mobile Ad Hoc Networks," in *Third Ann. Med-Hoc-Net. Workshop*, 2004.
- [5] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks", in *Proc. IEEE INFOCOM*, Mar. 2004.
- [6] G. Sharma, R. Mazumdar, and N. Shroff, "Delay and capacity trade-offs in mobile ad hoc networks: a global perspective", in *IEEE/ACM Trans. on Netw.*, vol. 15, no. 5, pp. 981-992, Oct. 2007.
- [7] 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.
- [8] L. Ying, S. Yang and R. Srikant. "Optimal Delay-Throughput Trade-Offs in Mobile Ad Hoc Networks", in *IEEE Trans. Inform. Theory*, Vol. 54, No. 9, pp. 4119-4143, Sept. 2008.
- [9] 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.
- [10] 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.

- [11] X. Li, S. Tang and O. Frieder, “Multicast Capacity for Large Scale Wireless Ad Hoc Networks,” in *Proc. ACM MobiCom 2004*, Sept. 2007.
- [12] X. Li, “Multicast Capacity of Wireless Ad Hoc Networks”, *IEEE/ACM Trans. Networking*, Vol.17, No. 3, pp. 950-961, Jan. 2008.
- [13] 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.
- [14] S. Zhou, L. Yin, “On Delay Constrained Multicast Capacity of Large-Scale Mobile Ad-Hoc Networks”, in *IEEE Infocom 2010*, San Diego, USA, Mar. 2010.
- [15] C. Hu, X. Wang, D. Nie and J. Zhao, “Multicast Scaling Laws with Hierarchical Cooperation”, in *IEEE Infocom 2010*, San Diego, USA, Mar. 2010.
- [16] A. Özgür and O. Lévêque, “Throughput-Delay Trade-Off for Hierarchical Cooperation in Ad Hoc Wireless Networks”, in *Proc. Int. Conf. Telecom.*, Jun. 2008.
- [17] 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.
- [18] 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.
- [19] R. Zheng and R. J. Barton, “Toward Optimal Data Aggregation in Random Wireless Sensor Networks”, in *IEEE Infocom 2007*, Anchorage , Alaska , USA, May 2007.
- [20] 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.