Throughput and Delay Analysis for Convergecast with MIMO in Wireless Networks

Luoyi Fu\textsuperscript{1,2}, Yi Qin\textsuperscript{1}, Xinbing Wang\textsuperscript{1}, Xue Liu\textsuperscript{3}
\textsuperscript{1}Depart. of Electronic Engineering, Shanghai Jiaotong University, China
\textsuperscript{2}The State Key Laboratory of Integrated Services Networks, Xidian University
\textsuperscript{3}School of Computer Science, McGill University
Email: \textsuperscript{1}\{yiluofu,qinyi\_33,xwang8\}@sjtu.edu.cn, \textsuperscript{3}xueliu@cs.mcgill.ca

Abstract—This paper investigates throughput and delay based on a traffic pattern, called converge-cast, where each of the \( n \) nodes in the network acts 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. We call them Convergimo Schemes. In static networks, our Convergimo 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 Convergimo scheme characterizes on joint transmission from multiple nodes to multiple receivers. With optimal network division where the number of nodes per cell is constantly 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 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 schemes are of interest not only from a theoretical perspective but also provide useful theoretical guidelines to future design of MIMO schemes in wireless networks.

Index Terms—Convergecast, Throughput, Delay, MIMO.

1 INTRODUCTION

Since the seminal work of Kumar [1] \textit{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 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 the packet has to endure a delay going to infinity. Due to the phenomenon that larger capacity is at the cost of a larger delay, some analysis on capacity-delay trade-offs 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 \textit{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 (\textit{[4]-[6]})) or transmit through coding or infrastructure support (\textit{[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 was well demonstrated by Aeron \textit{et al.} [10] who presented a MIMO collaborative strategy which achieves a per-node capacity of \( \Theta(1) \). Following that, Özgür \textit{et al.} [16] constructed a hierarchical cooperative scheme relying on distributed MIMO communications to achieve a linear capacity scaling. It turned out that nearly all the interferences can be canceled through hierarchical cooperation. Thereon, multicast scaling was taken into account in [15] under hierarchical cooperation which achieves an aggregate capacity of \( \Omega\left(\left(\frac{n}{\epsilon}\right)^{1-\epsilon}\right) \) for any \( \epsilon > 0 \). This also achieved a gain on capacity compared with previous works on multicast such as [11]- [14].

While the tradeoff for \textit{unicast} and \textit{multicast} traffic pattern have been extensively studied in previous work, \textit{converge-cast} is still a relatively new concept and under active research. \textit{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 multipath routes. Recently, there appeared many new applications such as real-time multimedia, battlefield com-
munications 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. Vast space of further improvement on 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”. 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. We call them Convgergimo schemes. For Convergimo scheme in a static network, 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. Through hierarchical operation, each cluster can be treated as a subnetwork and further divided into smaller clusters. 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, by transforming interfering signals into useful ones to the receivers, during hierarchical cooperative transmissions.

Under MANETs where hierarchical cooperation cannot be established due to the mobility of nodes, we devise another Convergimo 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 Convergimo 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 Convergimo scheme under MANETs, with optimal network division, the per-node throughput is $\Theta(1)$ with the corresponding delay reduced to $\Theta(k)$.
- Our results well unify and generalize some previous works since all of them can be easily applied to other traffic modes. 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 Convergimo 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 6.

## 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. Since the total number of nodes is $n$, there must be some sources shared among different converge-cast sessions. For each destination, it will receive distinctive packets from its $k$ sources. 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 $P$, which is assumed to be a constant1. The channel gain between two nodes $v_i$ and $v_j$ at time $t$ is given by:

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

(1)

where $d_{ij}$ is the distance between the nodes, $\theta_{ij}(t)$ is the random phase at time $t$, uniformly distributed in $[0, 2\pi)$.

1. The power budget indicates the transmitting limit in our model and the actual power consumption can be expressed as $Pf(\cdot)$, where $f(\cdot)$ is a function related to the parameters such as number of nodes, number of sources for each converge-cast session and etc. When we focus on the result of power consumption in order sense, we can simply remove the term $P$ from the expression.
2.2 Definitions

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: 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, denoted by $\lambda$, is defined as

$$\lambda = \lim_{t \to \infty} \frac{m(t)}{t}.$$ 

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>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Total number of nodes in the network.</td>
</tr>
<tr>
<td>$k$</td>
<td>The number of sources for each destination in the network.</td>
</tr>
<tr>
<td>$h$</td>
<td>The number of layers a network is divided into.</td>
</tr>
<tr>
<td>$i$</td>
<td>The $i$th layer of the network, where $1 \leq i \leq h$.</td>
</tr>
<tr>
<td>$n_i$</td>
<td>The number of nodes in the $i$th layer.</td>
</tr>
<tr>
<td>$k_i$</td>
<td>The number of sources for each destination node in the $i$th layer.</td>
</tr>
<tr>
<td>$n_{c_i}$</td>
<td>The number of clusters in the $i$th layer.</td>
</tr>
<tr>
<td>$M$</td>
<td>The average number of nodes in each cell.</td>
</tr>
</tbody>
</table>

3 Convergimo 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 Convergimo Scheme 1 under Static Networks

As is shown in [16], hierarchical cooperation can achieve better throughput scaling than classical multi-hop 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.

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. Take layer $i$ for example. There are totally $n_i$ nodes at this layer. Treating it as a whole network, it is divided into $n_{c_i}$ clusters with $n_{c_i}/n_{c_i}$ nodes located in each of them. At layer $i - 1$, each of those $n_{c_i}$ clusters is further regarded as a whole network with totally number of nodes of $n_{c_i}/n_{c_i}$. The network is further divided into $n_{c_{i-1}}$ clusters.

With recursion operations, 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. A whole view of the hierarchical structure of Convergimo Scheme 1 is shown in 1.1 in supplementary file.

Since the algorithm is similar at each layer but with different scale, we will present our recursive cooperative scheme 1 at a particular layer $i$. For a specific layer $i$, 

\[
\frac{0, 2\pi}{\theta_{ij}[t]} \text{ are i.i.d. random processes across all } i \\
\text{and } k, \text{ independent of each other. } G \text{ and the path loss } \alpha \text{ are } \geq 2 \text{ are assumed to be constants. Then, the signal received by node } v_i \at \text{ time } t \text{ can be expressed as }
\]

\[
Y_i[t] = \sum_{j \in T[t]} h_{ij}[t]X_j[t] + Z_i[t] + I_i[t],
\]

where $Y_i[t]$ is the signal received by node $v_i$ at time $t$, $T[t]$ 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.

Since stochastic analysis and optimization are not the main focus in this paper, in the following discussions, we will simplify the notation by suppressing the dependency of the channel gains on the time index $t$.

MIMO Technology: We adopt multiple-input and multiple-output, or MIMO Technology in this paper. In radio, MIMO represents the use of multiple antennas at both the transmitter and receiver to improve communication performance. It is one of several forms of smart antenna technology. MIMO technology has attracted attention in wireless communications, because it offers significant increases in data throughput and link range without additional bandwidth or transmit power.

Number of Antennas: 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.
the number of sources $k_i$ for a destination node can be expressed as $k_i = k / \prod_{k=i+1}^{k=h-1} n_{ek}$. Note that these $k_i$ sources are part of some $k$ source nodes in the original network. Moreover, at layer $i$, the scheme is divided into three steps 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 packets 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 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 $h$.

**Step 2. Multi-hop MIMO Transmissions:** We construct a converge-cast tree (CT) spanning from source nodes to its associated destination node. 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 conduct the three substeps presented below, spanning from source clusters $S_{ij}$ to their common destination clusters $D_i$. Here $1 \leq j \leq k_i$. Denote $P_i = \{S_{ij}, D_i, 1 \leq j \leq k_i\}$.

1) **Constructing the Euclidean spanning tree $\mathcal{E}_{EST}.$**
Firstly we divide the unit square into multi-level cells with side length $\frac{1}{2^{t-1}}$, where $t = \lfloor \log_2 k \rfloor$. 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 $\mathcal{E}_{EST} \rightarrow \mathcal{E}_{EST} \cup \{p_{ix}p_{iy}\}$ and $P \rightarrow P - \{p_{ij}\}$. Here $p_{ix}$ and $p_{iy}$ represents an edge connect by the two cluster $p_{ix}$ and $p_{iy}$. Subsequently we conduct this process by letting $y = t - 2, \ldots, 1, 0$.

2) **Getting the Manhattan routing tree $\mathcal{E}_{MRT}.$** For each edge $uv$ in $\mathcal{E}_{EST}$, 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, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{uv\}$, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{uv\}$, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{uv\}$.

3) **Obtaining the converge-cast tree for $P_i$, denoted as CT($P_i$).** For each edge $\overline{uv}$ in $\mathcal{E}_{MRT}$, we connect clusters cross by $\overline{vw}$ in sequence to form a path, denoted as $E(u,w)$. Then $\mathcal{E}_{CT} \rightarrow \mathcal{E}_{CT} \cup E(u,w)$, $\mathcal{E}_{MRT} \rightarrow \mathcal{E}_{MRT} \cup \{uv\}$. Finally, $\mathcal{E}_{CT}$ is the set of edges of CT($P_i$).

Note that the structure of both Euclidean spanning tree and Manhattan routing tree are invented from [10], based on a good approximation of a minimum connected dominating set (MCDS) of a random network. However, the direction and the amount of data flow in CT is different from those in [10]. Consider at layer $i$, there are $k_i$ sources distributed in $k_i$ source clusters. For any one of the $k_i$ source clusters, denoted by $j$, we assume there are $k_{ij}$ sources located in it. Obviously, $\sum_{j=1}^{k_i} k_{ij} = k_i$.

For a source cluster $j$ that has $k_{ij}$ sources, $k_{ij}/k_i$ nodes in this cluster participate in joint transmission of the packets to other clusters. Moreover, for a cluster denoted by $c_{ij}$, which will receive the data from its two adjacent clusters, denoted by $c_1$ and $c_2$, if $a/k_i$ of $c_i$ nodes are allowed for joint transmission in $c_1$ and $b/k_i$ of $c_2$ nodes are allowed for joint transmission in $c_2$, then $(a+b)/k_i$ of the nodes in $c_i$ will be active for joint transmission to next cluster. Under this transmission rule, when the data is finally flowed to the destination cluster, all the nodes in that cluster will be active for joint receiving.

In supplementary file, Figure 2 shows a simple example of the data flow on such 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 $\frac{t_i}{k_i}$ packets from other nodes, with each of them contributing $\frac{t_i}{n_{i-1}k_i}$ packets. Considering $n_{i-1}$ destinations in each cluster, the traffic load are $\frac{t_i}{n_{i-1}k_i}$ 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.

3.1.2 Throughput and Delay Analysis under Converging Scheme 1

Now we focus on throughput and delay that can be achieved under the scheme presented in 3.1.1. We first derive 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

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

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

**Proof:** Provided in 1.4 in supplementary file. \hfill \Box
Theorem 1: In static wireless networks, by adopting our Convergimo scheme 1, we can achieve an aggregate throughput of

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

with the delay of

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

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

Lemma 2: (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 3: Given $k_i$ independently and uniformly distributed source nodes in the network at layer $i$, the number of source clusters $c_i$ is given by

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

Proof: The proof is similar to that of Lemma 4.5 in [15] and we do not present the 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 [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 $\Theta \left( n_i^{a-1} k_i^{-1} \right)$ is achievable at layer $i-1$ w.h.p., where $0 \leq a < 1$, $-1 \leq b \leq 0$ and $a + b < 0$. It is easy to obtain that the total time to complete $k_i t_i$ traffic loads is

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

Hence, the throughput can be expressed as

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

Due to page limitations, we do not present the detailed proof here. Instead, we will show in 1.5 in supplementary file how to obtain the optimal throughput at the network at layer $h$, given $T_i$ in the above equation.

4 Convergimo Scheme under MANETs

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.

Remark 1: Although the i.i.d. mobility model may appear to be unrealistic, it has been widely adopted in the literature because of its mathematical tractability. Due to its property that the position of each node at different time slots and between different nodes are both independent, it will simplify the analysis to some extent. Note that other mobility models, such as the Brownian motion, random walk and random waypoint, possess the Markov property between time slots, which complicates the analysis.

4.2 Convergimo Scheme 2 under MANETs

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.

4.2.1 Convergimo 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.

- 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.
• 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.
• 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.

A simple illustration of our scheme is shown in Figure 3 in supplementary file.

4.3 Analysis of Throughput and Delay under Convergimo 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 Convergimo 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:

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

(9)

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 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 $M = \Theta \left( \frac{n}{\log(n)} \right)$ w.h.p. if $\lim_{n \to \infty} \frac{c}{\log(n)} = \infty$.

Proof: Provided in 1.6 in supplementary file.

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

Viewing from the perspective of network coding, the central problem arises: how long does it take for a destination node to receive at least $\Theta(k)$ combinations 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.

Lemma 7: 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

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

(10)

Proof: Provided in 1.7 in supplementary file.

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 have $\mathbb{E}[D_2] = \Theta(Mk)$. Therefore, the total delay achieved under our scheme is

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

(11)

4.3.2 Analysis on Throughput

Lemma 8: Under Convergimo scheme 2, we can achieve a per-node converge-cast throughput of

$$
\lambda = \begin{cases}
\Theta \left( \frac{1}{\log(n)} \right) & \text{if } k = o(\log(n)) \\
\Theta \left( \frac{\log(n)}{n^2} \right) & \text{if } k = \omega(\log(n))
\end{cases}
$$

(12)

in an MANET.

Proof: Provided in 1.10 in supplementary file.

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

5 Discussion

5.1 The advantage of Our Convergimo Schemes

In static network, our Convergimo 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. More specifically, we have shown in our analysis that to achieve the optimal throughput, the network division should be

$$
n_{i-1} = \Theta \left( \frac{k^{\frac{1}{4}}}{n^{\frac{1}{2}}} \right) \quad \text{if } n_c = O(k_i) \quad \text{and} \quad n_{i-1} = \Theta \left( (n_c k_i)^{\frac{1}{4-\kappa}} \right) \quad \text{if } n_c = \Omega(k_i).
$$

However, this optimal result is achieved under a given $h$. Varying $h$ will lead to different optimal network division. Our aggregate
5.2 Delay-throughput Tradeoff

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

\[
\begin{align*}
\Omega \left( n^{\frac{2k^2-4h+4}{2n-1} \cdot k^{-\frac{2n^2-2h}{2n-1}}} \right), & \quad \text{if } k = \Theta(n^{\frac{1}{m-2}}) \\
\Omega \left( n^{\frac{h^2-2h+1}{2n-1} \cdot k^2-\frac{2h^2-2h+1}{2n-1}} \right), & \quad \text{if } k = \Theta(1)
\end{align*}
\]  

(13)

MANETs: The delay/throughput tradeoff obtained under mobile network is \(M^2k\). It 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 Relationship with Other Traffic Patterns and Comparison with Previous Results

When applying Convergimo schemes in both static and mobile networks to them, we can get the throughput and delay of these traffic patterns, as are shown in Table 2. Then, we also make some comparison between our results with those provided in some previous works. The comparison is also shown in Table 2. The throughput in the table is unified to the per-node throughput. For static network, our results can achieve the similar performance to unicast presented in [16] and converge-cast \((k = n)\) in [17], [18], [19] and [20]. For MANETs, a gain of \(n\) is achieved on unicast throughput compared with that obtained in [7]. 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. The multicast and broadcast result is close to that of [7] with only \(\log n\) factor. Because 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.

<table>
<thead>
<tr>
<th>Network</th>
<th>Traffic</th>
<th>Throughput</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>Unicast</td>
<td>Our Convergimo Scheme 1: (\Theta(n^{\frac{2k^2-4h+4}{2n-1} \cdot k^{-\frac{2n^2-2h}{2n-1}}}))</td>
<td>(\Theta(n^{\frac{h^2-2h+1}{2n-1} \cdot k^2-\frac{2h^2-2h+1}{2n-1}}))</td>
</tr>
<tr>
<td></td>
<td>Conver-cast</td>
<td>Hierarchical Cooperation Scheme in [16]: (\Theta(\frac{n^{b}}{\log n})) ((0 &lt; b &lt; 1))</td>
<td>(\Theta(n^{\frac{1}{m-2}} \log n)) ((0 &lt; b &lt; 1))</td>
</tr>
<tr>
<td>Mobile</td>
<td>Unicast</td>
<td>Our Convergimo Scheme 1: (\Theta(1))</td>
<td>(\Theta(1))</td>
</tr>
<tr>
<td></td>
<td>Multicast</td>
<td>Our Convergimo Scheme 2: (\Theta(\frac{1}{\log(n)}))</td>
<td>(\Theta(\log(n)))</td>
</tr>
<tr>
<td></td>
<td>Broadcast</td>
<td>Our Convergimo Scheme 2: (\Theta(\frac{1}{\log(n)}))</td>
<td>(\Theta(\log(n)))</td>
</tr>
</tbody>
</table>

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.

Throughput result is obtained based on this optimal network partition at each layer with recursion operations from layer 1 to layer \(h\). Since our major concern is the throughput, the optimal \(h\) should be the one that maximizes the throughput. Given the number of hierarchical layers, the aggregate throughput in our paper is \(\Theta(n^{\frac{2k^2-4h+4}{2n-1} \cdot k^{-\frac{2n^2-2h}{2n-1}}})\). It can be seen from this equation that the throughput increases as \(h\) becomes large. When the number of layers \(h\) are sufficiently large in Convergimo 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.

6 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 \(\Theta(k)\).
ACKNOWLEDGEMENTS
This work is supported by The State Key Laboratory of Integrated Services Networks, Xidian University, National Basic research grant (2010CB31803), NSF China (No. 60832005); China Ministry of Education Fok Ying Tung Fund (No.122002); Qualcomm Research Grant; National Key Project of China (2009ZX03003-006-03, 2010ZX03003-001-01); National High tech grant of China (2009AA01Z248).

REFERENCES
Xue Liu  Dr. Liu obtained his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 2006. He obtained his B.S. degree in Mathematics and Master's degree in Automatic Control both from Tsinghua University, Beijing, China. He is now an Associate Professor in the Department of Computer Science & Engineering at University of Nebraska of Lincoln (UNL). He is also the Samuel R. Thompson Associate Professor in the College of Engineering.
1 Some illustrations of Our Convergimo Schemes

1.1 An Illustration of Hierarchical Structure of Convergimo Scheme 1

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 $h$. 
1.2 An Illustration of step 2 in Convergimo Scheme 1

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 of the nodes are allowed for transmission in the two source clusters on the left while 1/2 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.
1.3 An Illustration of Convergimo Scheme 2

![Diagram of cooperative scheme 2](image)

Figure 3: 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 sources-destination pairs in the network. (b) shows the case where there are no sources-destination pairs in the cell.

1.4 Proof of Lemma 1

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 $d_1, d_2, \ldots, d_k$) 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,

$$\sum_{i=1}^{k} \lambda_i \leq k \log \left( 1 + \frac{P}{N_0} \sum_{i=1, s_i \neq d}^{n} \frac{G_{s_i,d}^2}{d_{s_i,d}^2} \right). \quad (1)$$
According to Lemma 1 in [15], the distance between \( s_i \) and \( d \) is larger than \( \frac{1}{n^{1+\epsilon}} \) \( \text{w.h.p.} \) Thus,

\[
\sum_{i=1}^{k} \lambda_i \leq k \log \left( 1 + \frac{PG}{N_0} n_i^{a(1+\delta)+1} \right) \leq Ck \log n, \tag{2}
\]

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

\[
\lambda_i \leq C \log n, \tag{3}
\]

which implies that

\[
\sum_{i=1}^{n} \lambda_i \leq C n \log n. \tag{4}
\]

This completes our proof.

1.5 Proof of Throughput and Delay Provided in Theorem 1

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 3 and Lemma 4 in the paper, we have the following two cases:

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

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

\[
T_i = \widetilde{\Theta} \left( \frac{n_i n_{i-1}}{\sqrt{n_{i-1} k_{c_i} + n_{i-1}^{2-a-k^{-b}n_{i-1}^{b}}}} \right) = \widetilde{\Theta} \left( \frac{1}{n_{i-1}^{1-a-b} k_i^{1-b} n_i^{b}} \right). \tag{5}
\]

The result is optimized when \( T_i \) is maximized. Taking the derivative of \( T_i \) on \( n_{i-1} \) and letting \( \frac{dT_i}{dn_{i-1}} = 0 \), we can get \( n_{i-1} = \Theta \left( k_i \frac{n_{i-1}^{1-a-b} n_i^{b}}{k_i^{1-b} n_i^{a-b}} \right) \). Then, \( n_{c_i} = \frac{n_i}{n_{i-1}} = k_i^{\frac{1}{1-a-b}} n_i^{\frac{b}{1-b}} = O(k_i) \) and \( k_i = \Omega \left( \frac{n_i^{\frac{a-b}{1-b}}}{} \right) \).

At the bottom layer, the aggregate throughput is \( \frac{1}{n_i} \). 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 \( n_i = k_i^{\frac{1}{1-a-b}} n_{i-1}^{\frac{b}{1-b}} \) and \( T_i = k_i^{\frac{1}{1-a-b}} n_i^{\frac{b}{1-b/2}} \), the recursion calculation is listed as follows:
\[
\begin{array}{cccc}
i & a & b & n_i & T_i \\
2 & 0 & -1 & k_2^{1 \over 2} n_1^{1 \over 2} & k_2^{1 \over 4} n_2^{1 \over 2} \\
3 & 2 & -1 \over 3 & k_3^{1 \over 2} n_2^{1 \over 2} & k_3^{1 \over 4} n_3^{1 \over 2} \\
\end{array}
\]

\[h = \frac{2h-4}{2h-3} = \frac{1}{2h-3} \frac{1}{k_h^{\pi_n^{2,2}} n_h^{\pi_n^{2,2}} 2^{h-1} n_h^{\pi_n^{2,2}}} k_h^{1 \over \pi_n^{2,2}} n_h^{1 \over \pi_n^{2,2}} 2^{h-2} n_h^{1 \over \pi_n^{2,2}}\]

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

\[T = \Theta \left( \frac{n_{\pi_n^{2,2}}}{2} \cdot k^{-\frac{2n}{\pi_n^{2,2}}} \right).\]  

In the optimized result, \(k = \Omega \left( \frac{n_{\pi_n^{2,2}}}{2} \right) = \Omega \left( n_{\pi_n^{2,2}} \right)\).

Moreover, since \(n_i = k_i \pi_n^{2,2} n_{i-1} \pi_n^{2,2}\) and \(k_{i-1} = \Theta \left( \frac{k_i}{n_i} \right) = \Theta \left( \frac{k_i}{n_i} \cdot n_{i-1} \right)\), we have \(\frac{k_i}{n_i} = \Theta \left( \frac{h_i}{n_i} \right)\) and \(\left( \frac{k_i}{n_i} \right) \cdot n_i^{2i-3} = n_{i-1}^{2i-3}\). And this yields to the following derivation:

\[
n_{i-1} = \left( \frac{n_i}{k_i} \right)^{1 \over \pi_n^{2,2}} \cdot n_i^{2i-3} = \left( \frac{n_i}{k_i} \right)^{1 \over \pi_n^{2,2}} \cdot n_i^{2i-3} \\
= \left( \frac{n_i}{k_i} \right)^{1 \over \pi_n^{2,2}} \cdot \left( \frac{n_i}{k_i} \right)^{2i-3} \cdot n_{i+1}^{2i-3} \\
= \left( \frac{n_i}{k_i} \right)^{2i-3} \cdot n_{i+1}^{2i-3}.
\]

Hence,

\[
n_1 = \left( \frac{n_i}{k_i} \right)^{1 \over \pi_n^{2,2}} \cdot \frac{1}{2} \left( \frac{1}{2} + \frac{1}{2} + \frac{1}{2} - \frac{2n}{\pi_n^{2,2}} \cdot \frac{2n}{\pi_n^{2,2}} \right) n_{i-1}^{2i-3} \\
= \left( \frac{n_i}{k_i} \right)^{1 \over \pi_n^{2,2}} \cdot \frac{1}{2n_{i+1}^{2i-3}} n_{i-1}^{2i-3},
\]

from which we can obtain \(n_1 = n_{\pi_n^{2,2}} k^{-\frac{h-1}{\pi_n^{2,2}}}\).

Now we turn to the analysis on delay at layer \(i\), denoted as \(D(i, k)\). Through Equation (7), we get \(D(h, k) = \frac{n_B}{k}\), where \(B\) is minimum size of data transmitted at layer \(h\), i.e., \(B = \Theta \left( \prod_{i=1}^{h-1} \frac{n_i}{k_i+1} \right) = \Theta \left( \prod_{i=2}^{h-1} \frac{n_i}{k_i+1} \cdot \frac{n_1}{k} \right) = \Theta \left( \frac{2^{h-2} n_1}{k} \right) = (n) \frac{2^{h-2} n_2}{2h-4} k^{-\frac{2n}{\pi_n^{2,2}}}\).

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

\[D(h, k) = \Theta \left( n_{\pi_n^{2,2}} \cdot k^{-\frac{2n}{\pi_n^{2,2}}} \right).\]  

(7)
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:

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

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

1.6 Proof of Lemma 5

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, $\Pr[X_i = 1] = 1/c$.

According to Chernoff bounds, we have

$$\Pr \left\{ \sum_{i=1}^{n} X_i > (1 + \delta) \frac{n}{c} \right\} \leq e^{-\delta^2 \frac{n}{c}}$$  
and

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

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

$$\Pr \left\{ \sum_{i=1}^{n} X_i - \frac{n}{c} > \delta \frac{n}{c} \right\} \leq e^{-\theta^2 \frac{n}{c}} < e^{-\theta \frac{n}{c \log c}} \to 0$$  
as $n \to \infty$. Thus, $M = \Theta(\frac{n}{c})$ w.h.p. This completes our proof.

1.7 Proof of Lemma 7

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

$$\beta_{t+1} = \frac{J_t + a_1 + a_2 + \ldots + a_{J_t}}{J_t},$$  

(13)
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 \( \frac{\rho}{\rho_t} = \Omega(k) \), then the total number of cells where the \( k \) sources are located, denoted by \( k_c \), satisfies \( k_c = \Theta(k) \). In this case, initially, \( J_t \) is much smaller than \( \frac{\rho}{\rho_t} \). 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 \( \frac{\rho}{\rho_t} \), the number of cells where packet-holding nodes are located becomes the equal order of \( \frac{\rho}{\rho_t} \), which can guarantee that there are on average \( \Theta \left( \frac{M}{n} \right) \) such packet-holding nodes per cell per time slot.

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

Then, we have the following lemmas:

**Lemma 1**: 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

\[
\mathbb{E}[\mathcal{D}_1] = \Theta \left( \frac{\log \left( \frac{n}{k} \right)}{M' \log(1 + M')} + \Theta \left( 1 + \frac{1 + \log(n - \frac{\rho}{\rho_M})}{M - 1} \right) \right),
\]

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

**Proof 1**: Provided in Section 1.8.

**Lemma 2**: 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

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

**Proof 2**: 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

\[
\mathbb{E}[\mathcal{D}_1] = \Theta \left( 1 + \frac{n (1 + \log(n - k))}{M(n - k)} \right).
\]
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 $\mathbb{E}[D_1] = M \cdot \mathbb{E}\{\mathcal{P}_1\}$. Now we give the following lemma related to $\mathbb{E}[D_1]$.

**Lemma 3**: 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

$$\mathbb{E}[D_1] = \Theta(M) \quad \text{if} \quad M = \omega(\log(n))$$

$$\mathbb{E}[D_1] = \Theta(\log(n)) \quad \text{if} \quad M = o(\log(n)).$$

(17)

**Proof 3** Provided in Section 1.9.

### 1.8 Proof of Lemma *1*

For case 1, it is obvious that $\mathbb{E}\{a_t\} \approx M$. Then we have

$$\mathbb{E}\{\beta_{t+1}|J_t\} = \frac{J_t + J_t\mathbb{E}\{a_t\}}{J_t} \approx 1 + M. \quad (18)$$

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

$$\mathbb{E}\{J_t\} \approx k(1 + M)^t. \quad (19)$$

Note that the period ends when $J_t = \frac{n}{M}$. Denoting $D_{1_t}$ as the time required for $k$ packet-holding nodes to grow to $\frac{n}{M}$, then

$$\mathbb{P}[D_{1_t} > t|J_t] \geq \left(1 - \frac{M}{n}\right)^{tJ_t}. \quad (20)$$

Moreover, we have

$$\mathbb{E}\{D_{1_t}\} \geq t\mathbb{P}[D_{1_t} > t]$$

$$= t\mathbb{E}_{J_t}\{\mathbb{P}[D_{1_t} > t|J_t]\}$$

$$\geq t\mathbb{E}_{J_t}\left\{(1 - \frac{M}{n})^{tJ_t}\right\}$$

$$\geq t\left(1 - \frac{M}{n}\right)^{t\mathbb{E}_{J_t}}. \quad (21)$$

The above inequality holds for all $t > 0$. We choose $t$ to be a base $(1 + M)$ logarithm: $t \triangleq \frac{1}{M} \log_{1+M}\left(\alpha(\frac{n}{M})^{M^t}\right)$, 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 (21), we get

\[
\mathbb{E}\{D_1\} \geq \frac{\log(\alpha) + M^\delta \log(\frac{n}{k})}{M \log(1 + M)} \left\lfloor \left(1 - \frac{M}{n}\right)^n \right\rfloor \frac{k \left( \frac{n}{k} \right)^{\frac{t}{n \log(1 + M)}}}{n \log(1 + M)} \tag{22}
\]

Note that \( \left\lfloor \left(1 - \frac{M}{n}\right)^n \right\rfloor \xrightarrow{n \to \infty} e^{-1} \) when \( n \) goes to infinity. Now we consider the term \( \frac{k \left( \frac{n}{k} \right)^{\frac{t}{n \log(1 + M)}}}{n \log(1 + M)} \). Obviously,

\[
\frac{k \left( \frac{n}{k} \right)^{\frac{t}{n \log(1 + M)}}}{n \log(1 + M)} < \frac{(n/k)^{M-(1-\delta) \log(n/k)}}{n/k} \to 0
\]
as \( n \to \infty \). Hence, it follows that

\[
\left\lfloor \left(1 - \frac{M}{n}\right)^n \right\rfloor \xrightarrow{n \to \infty} 1.
\]

This implies

\[
\lim_{n \to \infty} \mathbb{E}\{D_1\} \geq \frac{\alpha \log \left( \frac{n}{k} \right)}{M^{1-\delta} \log(1 + M)}.
\tag{23}
\]

To obtain a tight bound on Inequality (23), we choose \( \delta \) to be a value very close to 1. Thus, it can be inferred that

\[
\mathbb{E}\{D_1\} = \Theta \left( \frac{\log \left( \frac{n}{k} \right)}{M^{1-\delta} \log(1 + M)} \right),
\tag{24}
\]

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

When \( J_t \) grows to \( \frac{n}{k} \), unlike the previous process, where \( \mathbb{E}\{a_t\} \) almost remains unchanged per time slot, the duplicate rate \( \mathbb{E}\{a_t\} \) starts to vary in different time slots. Let \( m \) denote the number of nodes which do not initially have the packet \( (m \leq n - \frac{n}{k}) \) and label these \( m \) nodes with \( \{x_1, x_2, \ldots, x_m\} \). Let \( X_t \) 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 \( \varphi > 1 - \left(1 - \frac{M}{n}\right)^{n-\frac{n}{k}} \geq 1 - e^{1-M} \).

At all times \( X_t \) are independent and identically distributed. Denoting \( D_{1z} \) as the time to expand the number of packet-holding nodes from \( \frac{n}{k} \) to \( n \), then the
random variable $D_{1,2}$ is equal to the maximum value of at most \( m = \left\lfloor \frac{n}{M} \right\rfloor \) i.i.d. variables. Hence, $\mathbb{E}[D_{1,2}] \leq \mathbb{E}\{\max\{X_1, X_2, \ldots, X_m\}\}$. Now we consider new random variables \( \{Y_1, Y_2, \ldots, Y_m\} \) 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, $\mathbb{E}[D_{1,2}] \leq 1 + \mathbb{E}\{\max\{Y_1, Y_2, \ldots, Y_m\}\}$.

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 $M + 1$ racing exponentials. It follows that

$$\mathbb{E}\{I_1 + I_2 + \ldots + I_m\} = \frac{1}{\nu} \sum_{j=1}^{m} \frac{1}{j}. \quad (25)$$

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,

$$\mathbb{E}[D_{1,2}] \leq 1 + \frac{1}{\nu}(1 + \log m) \leq 1 + \left(1 + \log\left(\frac{n - n/M}{M-1}\right)\right). \quad (26)$$

And

$$\mathbb{E}[\mathcal{D}_1] = \mathbb{E}[D_{1,1}] + \mathbb{E}[D_{1,2}] = \Theta\left(\frac{\log\left(\frac{n}{M}\right)}{M \log(1+M)}\right) + \Theta\left(1 + \frac{1 + \log(n - n/M)}{M-1}\right). \quad (27)$$

### 1.9 Proof of Lemma *3*

For case 1 presented in Lemma 7, we can get

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

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

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

10
For case 2 presented in Lemma 7, we have

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

Since \( k = o(n) \), \( \frac{n(1 + \log(n - k))}{n - k} \approx \log(n) \). 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

\[ \mathbb{E}[D_1] = \begin{cases} 
\Theta(M) & \text{if } M = \omega(\log(n)) \\
\Theta(\log(n)) & \text{if } M = o(\log(n)) 
\end{cases}. \tag{28} \]

This completes our proof of Lemma *3.

1.10 Proof of Lemma 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 (9).