Wireless Device-to-device Caching Networks with Distributed MIMO and Hierarchical Cooperations

Jiajia Guo[1,3], Jinhong Yuan[1], and Jian A. Zhang[2]
[1]University of New South Wales, Australia, [2]University of Technology Sydney, Australia, [3]Data61, CSIRO, Australia
Email: jiajia.guo@students.unsw.edu.au; j.yuan@unsw.edu.au; Andrew.Zhang@uts.edu.au.

***Abstract*—In this paper, we propose a new caching scheme for a random wireless device-to-device (D2D) network of □ nodes with local caches, where each node intends to download files from a prefixed library via D2D links. Our proposed caching delivery includes two stages, employing distributed MIMO and**

hierarchical cooperations respectively. The distributed MIMO is applied to the first stage between source nodes and neighbours of the destination node. The induced multiplexing gain and diversity gain increase the number of simultaneous transmissions, improving the throughput of the network. The hierarchical cooperations are applied to the second stage to facilitate the transmissions between the destination node and its neighbours. The two stages together exploit spatial degrees of freedom as well as spatial reuse. We develop an uncoded random caching placement strategy to serve this cooperative caching delivery. Analytical results show that the average aggregate throughput of the network scales almost linearly with □, with a vanishing outage probability.

***Index Terms*—Scaling law, caching, distributed MIMO, device-to-device communications**

I. Introduction

Among many proposals for future wireless networks, wireless caching has been proposed as a cost-effective way to handle the high traffic requirement caused by content delivery applications, especially on-demand video streaming [1]–[3]. The conventional ad-hoc network model in [4] considers □

nodes randomly distributed in a unit area and arbitrarily grouped into source-destination pairs. The result in [4] show that for such a network, a multihop strategy is capable of achieving an aggregate throughput scaling of $\Theta\left(\sqrt{\square}\right)$. Note

that this result is achieved based on a *protocol model*, where nodes within a certain distance are able to communicate and only one node is allowed to transmit at a time within a radius. Beyond the protocol model, a *physical model* was introduced in [5] to take into consideration wireless channel properties such as pathloss, fading, AWGN noise and interference. Based on this model, when the network is dense (fixed area with node density growing), a hierarchical cooperation scheme proposed in [5] achieves a near-linear throughput scaling of $\Theta\left(\square^{1-\in}\right)$ where $\in$ can be arbitrarily small. Moving to the

studies of wireless caching, a similar caching network has been studied in recent works [6], [7]. The caching network consists of □ nodes randomly distributed in a unit area, while each node is equipped with a local cache of □ files and requests files from a library of □ files $(\square \geq \square)$ according to a priori popularity distribution. Following the protocol

model, the network is divided into clusters and a user can only communicate inside its own cluster. They showed that for a single-hop cache network, with an optimized caching placement strategy and cluster size, the network throughput scales as $\Theta\left(\frac{\square\square}{\square}\right)$, growing linearly with the size of local

$\sqrt{\quad})$

(caches □ . They later improved this scaling to Θ □

□

in [8], [9] by adopting the multihop strategy, where nodes get served by multi-relaying through the network.

Studies in [6]–[9] are based on the protocol model in [4]. We notice that the work in [3] uses the physical model and enlarges the network throughput by employing MDS code for the cached content to create cache-induced opportunis- tic CoMP. However, we are further interested in whether hierarchical cooperations in caching networks can achieve a better throughput scaling than the multihop scheme as in conventional ad-hoc networks. To address these questions, we investigate the caching network based on the physical model as well as considering hierarchical cooperations.

Another motivation of our work is the concern about the limited cache size in practice. We notice that compared with the vast library that users may request, the local cache size at each node is rather limited in realistic networks. Especially, in a D2D network, the cache size of a single device cannot be sufficiently large in many cases. For example, consider the on-demand video streaming case, the file library could be up to 1000 TB while the available cache size of a user device is usually less than 1 TB. In addition, users usually are willing to contribute only a small fraction of their local caches. This conscious motivates us to design a caching scheme for the small cache case (small □), where the network throughput will be poor according to existing works. Thus, in this paper, our primary focus is on the small cache case when designing the caching scheme. More specifically, our focus is not on whether a better throughput scaling with the size of caches can be achieved. Instead, we mainly investigate if a good scaling with the number of users can be achieved, even when the cache size compared with the size of library is small.

In this paper, we consider a decentralized D2D cached network similar to that in [6]–[9], where each node in the network has access to a size-limited local cache and would like to download files from a library through D2D links. We summarize our contributions as follows.

- We propose a new caching scheme employing distributed multiple-input and multiple-output (MIMO) transmissions and hierarchical cooperations for the considered

D2D network. Different from the protocol model widely used in D2D literatures [6]–[9], our proposed scheme is based on *the physical model* [5], [10] considering wireless channel properties such as pathloss and interference. The distributed MIMO technology is used in the cache delivery phase utilizing the "overheard" signals introduced by the physical model. Hierarchical cooperations are used to facilitate the transmissions between neighbours and the destination node. This design exploits spatial degrees of freedom in addition to spatial reuse.

- A random independent caching placement strategy is proposed to serve the proposed caching delivery, and an asymptotic expression of the cache-hit probability is derived. In our design, the files in the library are divided into packets and each node caches uncoded packets from different files, which lays a foundation for employing the distributed MIMO technique in the caching delivery phase.
- We derive the throughput scaling law for our proposed scheme. Our analysis shows that the average aggregate throughput of the network scales almost linearly with the number of nodes □, which outperforms the current

$\sqrt{\ }$)

scaling of $\Theta\left(\frac{\square}{\square}\square\right)$ and $\Theta(\square)$ □ when the local cache size is limited. We also prove that the outage probability approaches zero as □ goes sufficiently large.

Throughout this paper, we say that event □ happens with high probability if $\lim_{\square\to\infty} \Pr\{\square\} = 1$, and that event □ happens with a vanishing probability if $\lim_{\square\to\infty} \Pr\{\square\} = 0$. For two given functions □ (□) and □ (□), we say that □ (□) = Θ(□ (□)) if ∃ □₁, □₂ > 0, ∃ □₀, ∀ □ > □₀, □₁□ (□) ≤ □ (□) ≤ □₂□ (□).

II. System Model and Problem Formulation

Consider a network with □ wireless nodes, which are uniformly and independently distributed in a unit square. Each node □, □ ∈ □ = {1, · · · □}, has an average transmit power of □ watts and a local cache size of □ files. The size of caches in this paper is represented by the number of standard files (□ bits per file). D2D communications between nodes are considered in this network. A library with □ files is denoted by ℱ, ℱ = {□₁, · · · , □□}. Each node intends to download its requested file from the library ℱ through this cached

□□. Then, one realization of all nodes' requests, denoted by □, can be represented by

$$\square = \{\square_1, \cdots, \square_\square\}. \qquad (1)$$

In this paper, we consider that for any node in the network, its requested file index □□ is uniformly distributed over {1, · · · , □}. This stands as the worst case scenario where users' requests are quite scattered rather than gathered. This assumption is also adopted in [11] and fits the spirit in [6], [7] considering a "heavy tail" Zipf request distribution.

*Definition 3 (**Transmission policy**):* A transmission policy Π□ is a rule of designing and scheduling the D2D transmissions in the delivery phase. Generally, it includes two parts: one is the transmission protocol □ describing the feasible D2D links in the network, such that these D2D links can provide reliable communications, considering the physical constraints such as power and interference; and the other is the transmission scheduling □ describing the activated D2D links at one time.

Denote by □ a set of simultaneous transmission links.

*Definition 4 (**Average aggregate throughput**):* For a given caching placement □, node requests □ and a set of transmission links □, define □□ (□, □, □) as the number of useful received information bits per time-slot by node □ during one delivery phase. Adding □□ (□, □, □) from all nodes together, we have the aggregate throughput of the network as

$$\square_\square(\square, \square, \square) = \sum_{\square\in\square} \square_\square(\square, \tilde{\square}, \square). \qquad (2)$$

Let □ denote the *average aggregate throughput*, and

$$\square_\square = \square_{\square}\left[\square_{\tilde{\square}}\left[\square(\square, \square, \square)\right]\right], \qquad (3)$$

D2D network. A caching scheme □ is performed in two phases: caching placement phase and caching delivery phase.

*Definition 1 (**Caching placement strategy**):* A cache placement strategy Π□ is a rule to assign files from the library ℱ to where $\square_{\square}$ means averaging over the randomness of nodes' actual requests and $\square_{\square}$ means averaging over the randomness caused by scheduling during all delivery phases. We see that $\square_{\square}$ is a function of the caching placement strategy $\Pi_{\square}$ and the transmission policy $\Pi_{\square}$.

*Definition 5 (**Outage probability**):* For given □, □ and □, if a node's request cannot be satisfied through the delivery phase, we say that this node is in outage. Denote the a node's local cache in the caching placement phase. Let $\mathcal{M}_{\square}$ represent the local storage of node □. Then, a particular cache placement $\square_{\square}$ at node □ can be viewed as a mapping from the library $\mathcal{F}$ to the local memory $\mathcal{M}_{\square}$. One realization of caching placement at all nodes is denoted by $\square \triangleq \{\square_{\square}, \square \in \square\}$. Note that the caching placement is done without a priori knowledge of the nodes' actual requests.

*Definition 2 (**Users' request**):* At each request time, each node makes a request to a file $\square_{\square}$, $\square_{\square} \in \mathcal{F}$, randomly and independently. Denote the index of node □'s requested file by

number of nodes in outage by $\square_{\square}(\square, \square, \square)$ in one delivery phase, i.e.,

$$\square_{\square}(\square, \square, \square) = \sum_{\square} \mathbf{1}(\square_{\square}(\square, \square, \square) = 0). \quad (4)$$

Define the *outage probability* of the network as

$$\square_{\square} \overset{\square\,\square}{=} \frac{\square_{\square}[\square_{\square}[\square_{\square}(\square, \square, \square)]]}{\square}, \quad (5)$$

where $\square_{\square}$ and $\square_{\square}$ are described under (3).

In this paper, we focus on the scaling of the average aggregate throughput $\square_{\square}$ with an increasing number of □, while the power constraint of each node and the occupied spacial area of all nodes remain constant. In addition, we will assume that the size of the library □ and the size of local cache □ increase with □, i.e.,

$$\square = \square^{\square} \text{ and } \square = \square^{\square}, \quad (6)$$

where □ 0 and □ 0. We further assume that $0 < □ - □ 1$, meaning the total memory size of the network □ □ is larger than the library size □ while the local cache at each node cannot contain the whole library.

Let $□_□ = \Theta(□^□)$, $□ \geq 0$. The goal of this paper is to jointly optimize the caching strategy $\Pi_□$ and the transmission policy $\Pi_□$ so that the exponent □ is maximized while the outage probability is vanishing, i.e.,

$$\mathrm{p}: \max_{\Pi_□,\Pi_□} □$$

$$□.□.\ \lim_{□\to\infty} □_{□□□} = 0. \qquad (7)$$

III. Main Results

We introduce an achievable scaling law when the number of nodes □ is sufficiently large as follows.

Theorem 1: For the considered caching wireless network with a library of □ files and local caches of □ files, if □ □ > □, as $□ \to \infty$, the following scaling law is achievable with high probability:

$$□_□ = \Theta\left(□^{\frac{□}{□+1}}\right), \qquad (8)$$

where □, $□ \geq 1$, is an integer constant independent of □.

Proof. An achievable scheme is presented in Section IV. The choice of the parameter □ will also be discussed there. ■

Remark 1: From (8), with □ increases, $\frac{□}{□+} \to 1$. This indicates that with the help of caching, the average throughput of the network can increase almost linearly with the number of nodes □. Note that the scaling $\Theta\left(□^{\frac{□}{□}}\right)$ in this paper is different from the scaling $\Theta\left(\frac{□□}{□}\right)$ in [6] [11] or $\Theta\left(\sqrt{\frac{□}{□}}\right)$ in [8] [9]. Fig. 1 compares the throughput scaling laws of the caching network in [6] [11], [8] [9] and Theorem 1. Recall that the most interesting region is that □ □ ≥ □, i.e., $0 < □ - □ \leq 1$. We see that $1 \leq \frac{□□}{□} < \sqrt{\frac{□}{□}} < □$, $□ \leq □$ and in our result $□ \leq □^{\frac{□}{□+1}} < □$ (since $□ \geq 1$). This is shown in the figure where these three scalings exhibit different throughput ranges in the region of $0 < □ - □ \leq 1$. It is also immediate to see that the scalings in [6] [11] and [8] [9] are both functions of $□ - □$, the relative exponent of □

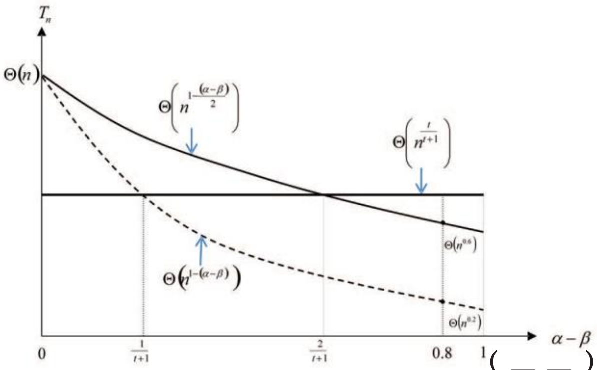


Fig. 1: Achievable throughput scaling laws of $\Theta\left(\frac{□□}{□}\right)$ in [6] [11], $\Theta\left(\sqrt{\frac{□}{□}}\right)$ in [8] [9] and $\Theta\left(□^{\frac{□}{□+1}}\right)$ in (8) respectively.

each file in the library is partitioned into □ packets of equal size. More specifically, we partition a file $□_□ \in \mathcal{F}$ into □ packets of equal size $□_{□,□}$, $□ \in \{1, \cdots, □\}$, i.e.,

$$□_□ = \bigcup_{□\in\{1,\dots,□\}} □_{□,□}. \qquad (9)$$

Thus, there are □ □ packets in the library. Note that □ is an important parameter to be determined.

During the cache placement, each node fills in its local cache in an i.i.d. manner according to a same placement strategy $\Pi_□$. Each node randomly chooses □ □ packets from the □ □ packets of the library and stores them in its own cache. Denote the □th packet in node □'s cache $\mathcal{M}_□$ by $□^□$, we have

$$\mathcal{M}_□ = \bigcup_{□\in[1,□□]} □_□. \qquad (10)$$

The caching placement process at each node is described as follows. First, each node □ decides which files to choose from, by randomly generating □ □ file indexes $\{□^□\}$, $□ \in \{1, \cdots, □□\}$, $□^□ \in \{1, \cdots, □\}$. We use the following rule between the library size □ and the cache size □, however the scaling in Theorem 1 is irrelevant with $□ - □$. Furthermore, it is observed to determine the chosen file indexes for node □.

1) If the number of packets stored in each node □ □ is less than the number of files □

that our scaling performs better under a cache-size limited situation (with □ − □ large). For example, consider a caching network with □ = 1 and □ = 0.2. This setup corresponds to

a disadvantageous caching case where there are many files in the library however the local cache at each node is relatively small. For example, when □ = 10^8, there are 10^8 files in the library while each node only has a cache size of 40 files. The

aggregate average throughput in [6] [11] and [8] [9] will be

$\Theta\left(□^{0.2}\right)$ and $\Theta\left(□^{0.6}\right)$ respectively, while our scheme achieves $\Theta\left(□^{0.75}\right)$ when a typical value □ = 3 is applied to (8).

IV. An Achievable Scheme

A. Caching placement phase

In this paper, we propose an uncoded, distributed and randomized placement strategy $\Pi_□$. Before the cache placement,

(i.e., □ □ ≤ □), the packets for each node cache should be chosen from different files. That is, ∀ $□_1, □_2$ ∈ [1, □ □], $□^{□_1}$ ≠ $□^{□_2}$ can be □ . This recognized as a similar process of blindly drawing □ balls from a box of □ different balls at a □ time.

2) If □ □ > □, the file indexes are generated as follows. First, all file indexes 1, · · · , □ are chosen for □ = ⌊□ □⌋ times. That is, for □ ∈ [1, □], ∀□ ∈ [(□ − 1) □ + 1, □ □], $□^□$ = □ −(□ − 1) □. Then, for the rest of (□ □ − □ □) packets, their file indexes must be different. That is, ∀ $□_1, □_2$ ∈ [□ □ + 1, □ □], $□^{□_1}$ ≠ $□^{□_2}$.

After the file indexes are determined, node □ randomly picks up one of the packets from file □ with probability 1/□ and fills it in the local cache. Denote by $\mathcal{M}_□^□$ the □th chosen packet by node □ in file, we have that

$$□_□^□ = □_{□,□}, \text{ and } \mathcal{M}_□ = \bigcup_{□\in[1,□]} □_{□,□}. \quad (11)$$

These two steps ensure that the packets stored at one node come from different files to the most extent, which facilitates the delivery phase to be discussed next.

B. Caching delivery phase

1) Preliminaries:

a) Channel model: We use the line-of-sight *physical model* in [5], [10]. Denote the channel coefficient from node □ to node □ by $\boldsymbol{h}_{□□}$, then

$$\boldsymbol{h}_{□□} = \sqrt{□_{□}}\,(□_{□□})^{-\frac{□}{2}} \exp(□□_{□□}), \qquad (12)$$

where $□_{□□}$ is the distance between node □ to node □, $□_{□□}$ is the random phase, uniformly distributed in $[0, 2□)$, $□_{□}$ is the antenna gain and □ is the pathloss exponent of the environment. We have $□ \geq 2$ for the far-field assumption.

b) User clustering: Divide the entire network into square cells of area $□_{□}$ as shown in Fig. 2a. We call each square cell a *cluster* in this paper. Let $□_{□} = □^{-□}$, □ 0.We see that □ is an important parameter determining the cluster size, which will be chosen later. From literatures [4] [8], we know that the number of nodes in one cluster is $□_{□} = \Theta(□^{1-□})$.

We denote by □ the cluster that the destination node belongs to and by □ all the neighboring clusters of □.

2) Stage I: Distributed MIMO: We first consider the transmission policy for serving a single node, and then extend it to serving all nodes in the network.

Recall that node □ requires one file $□_{□_□}$ with □ packets. In order to collect all □ packets of file $□_{□_□}$, we need to determine □ source nodes, that store the □ packets in their caches respectively. Then, to serve node □, a virtual multiplexing MIMO transmission can be formed from these □ source nodes to the destination cluster □. For any requested file $□_{□_□}$ of node □, □ source nodes, $□_1^{□}, \cdots, □_{□}^{□} \in □$, are selected according to the following criteria.

1) Each of the source nodes has one different packet of the requested file $□_{□_□}$ in its local cache, i.e.,

$$\exists\, □_1, \cdots, □_{□} \in [1, □□], \; □^{□_1}_{□_1} \cup □^{□_2}_{□_2} \cup \cdots \cup □^{□}_{□} = □_{□_□}. \qquad (13)$$

2) All the source nodes cannot lie in the neighboring clusters of node □, i.e.,

$$□_1^{□}, \cdots, □_{□}^{□} \notin □. \qquad (14)$$

Note that if there are multiple candidates of source nodes,

So far, we have described the transmission policy $\Pi_□$ for serving a single node. For serving all nodes in the network, the same transmission policy $\Pi_□$ is applied in a time-division manner.

3) Stage II: Hierarchical Cooperations: The goal of this stage is to collect and jointly process the MIMO observations from Stage I at the actual destination node □. To this end, we apply hierarchical operations similar to that in [12] [13] as follows.

Within a destination cluster, each node quantizes the received signals and sends them to the destination node. The destination node then jointly processes the $□_{□}$ copies of superimposed signals received from previous multiplexing transmissions nodes. Thus an MIMO transmission from the source nodes to the destination node is formed through the two-stage cooperations. Note that the quantization does not change the linear scaling (of the number of independent transmitting streams) of MIMO capacity, which is proven in [5].

We notice that each node in the cluster wants to send independent messages to all other nodes in the same cluster. This communication problem is referred to as a *network multiple access problem* in [12] [13]. For this problem, a hierarchical three-phase cooperative transmission was proposed employing a similar idea of clustering, distributed MIMO and quantize-and-forward [12]. A hierarchical scheme with an improved scheduling, referred to as Method 4 in [13], achieves the best scaling and transmission rate by far to our best knowledge. We adopt this design for the network multiple access problem in Stage II to achieve the scaling.

Here, we discuss the case where some of the source nodes are in the destination cluster □. If some packets of the requested file $□_{□_□}$, say □ packets, can be found within cluster □, in Stage I, the number of transmitted packets will be $(□_{□} - □_{□})$. Thus, we can reduce the number of receivers to $(□_{□} - □_{□})$ and the consumed time slots for MIMO transmissions in Stage I remain unchanged. In Stage II, the number of transmitted packets to node □ will remain as $(□_{□} - 1)$. This is

we can randomly pick up any □ candidates that meet the criteria above. The criteria allow a destination node to choose source nodes from the entire network except for the neighbouring clusters □.

because node □ will only need to receive MIMO observations from $(\square_{\square} - \square_{\square} - 1)$ nodes, in addition to receiving the □□ packets from its neighbours. It is convenient to arrange that the neighbours with □□ packets are not scheduled as receivers in MIMO transmissions. Thus, Stage II can still

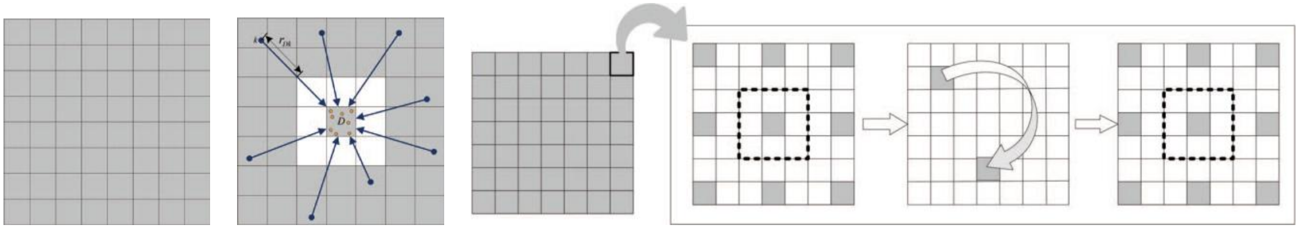
After selecting the source nodes $\square^{\square}_{1}$, . . . , □□, □ as transmitters, we choose all the □ nodes in the destination cluster □ □ as receivers. Each packet of file □□□ is simultaneously sent from the □ transmitters to the □□ receivers in a multiplexing way as shown in Fig. 2b. An MIMO transmission will be formed as long as the observations at all receivers can be jointly processed. We will discuss how to realize this joint processing in Stage II

be seen as a network multiple access problem with □□ nodes and it will also occupy the same time-slots. As a result, we see that in terms of the aggregate throughput, the case where some packets can be found within cluster □ is equivalent to the case where no source nodes are in cluster □ .

C. Aggregate throughput of the proposed scheme

We calculate the aggregate throughput for the proposed scheme as follows.

1) Required time-slots for Stage I: In order to guarantee reliable transmissions from the source nodes to the destination cluster, it is required that the number of independent transmitting streams □ should be no more than the number of receiving



(a) Clustering: Divide the entire network into square cells of area □ □ where □ □ = □ −□ , □ 0. □ determines the cluster size. The bigger □ is, the smaller each cluster is.

(b) Stage I: Distributed MIMO. The middle grey cluster represents the current destination cluster □. The surrounding grey area contains all candidates of source nodes.

(c) Stage II: Hierarchical Cooperations. All clusters work simultaneously. Within one cluster, a three-phase hierarchical cooperation is employed including clustering, distributed MIMO and quantize-and-forward. A time-division protocol is used for the first and third phases. For example, in this figure, only one cluster is allowed to transmit among 9 clusters (□ = 3).

Fig. 2: Caching delivery phase of the proposed scheme: (a) Clustering; (b) Stage I: Distributed MIMO; (c) Stage II: Hierarchical Cooperations.

antennas □ □, i.e.,

$$□ \leq □_□. \quad (15)$$

Let the geographical distance between transmitter □ and destination cluster □ be □ □ □ as shown in Fig. 2b. We use the following transmit power control mechanism such that

$$\mathbf{E}\left[|□_□|^2\right] = \frac{[}{□}(□_{□□})^□. \quad (16)$$

We see that the transmit power control mechanism is similar to that in [5] [12]. We then obtain the following lemma.[1]

Lemma 1: Under the channel model described in (12), applying the transmit power control in (16), a long-distance distributed MIMO transmission from the source nodes to the destination cluster □ achieves an aggregate rate scaling linearly with the number of transmitters □.

We see that the transmit power constraint at each node is satisfied in our design. This is because when serving one node, each node only needs to transmit $\frac{□}{[}(□_{□□})^□$ watts and only a fraction of $\frac{[}{□}$ nodes are transmitting. So in order to serve all □ nodes in the network, on average, each node needs a total transmit power

$$\frac{[}{[}(□_{□□})^□ \cdot \frac{[}{□} \cdot □ = □ \,(□_{□□})^{[} \leq \frac{□}{□} \quad (17)$$

which satisfies the transmit power constraint.

Based on our designs of Stage I, each packet has $\frac{□}{[}$ bits

where □ is the reuse factor, meaning that each cluster has one transmission opportunity every □ ^{2} time-slots. By substituting the number of nodes □ by □ □ in our problem, we obtain the total time □ □ □ for completing transmissions for a cluster of □ nodes as

$$□_{□_□} = \frac{□^2}{□□□□} □^{\frac{□}{□-1}}_2 □_□^{\frac{□}{□+1}}. \quad (19)$$

The optimal number of hierarchical stages □ for maximizing the throughput can be found by an exhaustive search for a given number of nodes □ □ according to [13].

3) Throughput scaling law of the proposed scheme: From previous designs for both stages, we know that □ □ bits in total are delivered to their destinations in $\frac{[}{□} + □_□$ time slots. Thus, the aggregate throughput is

$$□_□ = \frac{□□}{\frac{[}{□} + □_□} = \frac{□□}{□□ + □^2 □^{\frac{□}{□-1}} □_□^{\frac{□}{□+1}}}. \quad (20)$$

It is clear that in order to maximize □ □, the number of nodes in one cell □ □ should be as small as possible. Since □ ≤ □ □, the optimal □* □ should be

$$□^*_□ = □ \quad (21)$$

Remark 2: From (21), we see that the optimal number of nodes in one cluster $□_□$ and the number of $□$ packets of a file and all the $□$ packets of a file are simultaneously transmitted in a multiplexing way. Using Lemma 1, for any requested file of one node, a reliable transmission from source nodes to the destination cluster will take $□$ time slots. Therefore, in total, $\frac{□}{□}$ $□$ $□$ time slots are needed for serving all $□$ nodes in Stage I.

2) Required time-slots for Stage II: In this stage, we use hierarchical cooperations within one cluster. Recalling the results in [13], the time to complete the network multi-access transmission for a network of $□$ nodes with $□$ hierarchical stages and $□$-bit quantization is given by

$$□_□ = □ □^2 □^{\frac{□-1}{2}\frac{□+1}{□}}_{□}, \tag{18}$$

are coupled. On one hand, if choosing a large cluster size (a large $□_□$), the number of packets of a file $□$ should also be large. This is essentially attributed to our design in Stage I: with large clusters (a large $□_□$), the number of independent transmitting streams $□$ should also be large in order to utilize the multiplexing gain to the most extent; On the other hand, with a large $□$, $□_□$ must be large enough to support reliable multiplexing transmissions in Stage I.

Then, with (21), we get that

$$□_□ \underset{1}{=} \frac{□ □}{□ + □ □^2 □^{\frac{□}{2}} \, □} \overset{□+1}{=} \frac{□}{(\frac{□}{□} \, \frac{□}{+1}) \, □ + \Theta(□^{□})}. \tag{22}$$

By maximizing $□_□$ over the number of packets $□$, we obtain

$$□^* = \Theta\left(□^{\frac{□+1}{□}}\right) \tag{23}$$

[1] The proofs for Lemma 1 and Theorem 2 are omitted due to page limit. Detailed proof can be found in the journal version in [14].

when $\square^{*} = \Theta\left(\frac{\square}{\square}\ \square^{+1}\right)$.

It is interesting to see that the optimal throughput scaling $\square^{*}_{\square}$ and the corresponding $\square^{*}$ are on the same order, i.e.,

$$\square^{*}_{\square} = \Theta(\square). \tag{24}$$

This is because the improvement of the throughput from $\Theta(1)$ to $\Theta\left(\frac{\square}{\square}\right)$ is mainly attributed to the multiplexing gain achieved by distributed MIMO. For instance, a smaller □ means that one file is divided into fewer packets, and the number of simultaneous transmissions is reduced, which decreases the throughput $\square_{\square}$.

Furthermore, using the above results, we can also determine the cluster size in our design as follows. We see that to achieve the optimal throughput, $\square^{*}_{\square} = \square$ and $\square^{*}_{\square} = \Theta\left(\frac{\square}{\square}\ \square^{+1}\right)$ must be satisfied at the same time. Recall that $\square_{\square} = \Theta\left(\square^{1-\square}\right)$, □ must satisfy that $\Theta\left(\frac{\square}{\square}\right) = \Theta\left(\square^{1-\square}\right)$, i.e., $\square = \frac{1}{\square+}$.

D. Outage probability of the proposed scheme

We now prove that the throughput can be achieved with a vanishing outage probability as $\square \to \infty$.

Theorem 2: As $\square \to \infty$, the outage probability in our scheme is upper bounded by

$$\square_{\square\square\square} \le 1 - \square^{-\square\ \square\cdot\square}\ \square^{-\frac{(\square-(\square^{2}-1)\square\square)\square}{\square}}. \tag{25}$$

According to (25), We see that $\square_{\square\square\square}$ increases with the number of nodes in one cluster $\square_{\square}$. Recall that $\square_{\square} \ge \square$. Therefore, to achieve a small outage probability $\square_{\square\square\square}$, it is required that $\square^{*} = \square$. This agrees with the requirement for maximizing $\square_{\square}$ in (21). Thus, we have

$$\square_{\square\square\square} \le 1 - \square^{-\square\ \square\cdot\square}\ \square^{-\frac{(\square-(\square^{2}-1)\square)\square}{\square}}. \tag{26}$$

To see more clearly about the scaling of $\square_{\square\square\square}$ with □, we further extend (26) as follows. It is easy to verify that as $\square \to \infty$, $\frac{\square-(\square^{2}\square}{(\square} \to 1$. Therefore, $\square - (\square^{2}-1)\,\square$ can be approximated by □ when □ goes sufficiently large. Recall that $\square^{*}_{\square} = \Theta\left(\frac{\square}{\square}\ \square^{+1}\right)$, then we have

$$\square_{\square\square\square} \le 1 - \square^{-\square\ \square}\ \square^{-\frac{\square}{\square}}\ \frac{\square}{+1}, \tag{27}$$

where

V. Conclusions

We have investigated the throughput scaling problem in a wireless D2D network where each node is equipped with a local cache and would like to download files from a pre-fixed library. We apply distributed MIMO between source nodes and the neighbours of the destination node to increase the number of simultaneous transmissions in the network. We use hierarchical cooperations to provide a high backhaul capacity. We also establish an uncoded random caching placement strategy. Our analytical results show that the average aggregate throughput of the proposed scheme scales almost linearly with □, with a vanishing outage probability.

References

[1] K. Poularakis, G. Iosifidis, V. Sourlas and L. Tassiulas, “Exploiting Caching and Multicast for 5G Wireless Networks,” *Wireless Communications, IEEE Transactions on*, vol.PP, no.99, pp.1-1, 2016.

[2] M. Tao, E. Chen, H. Zhou and W. Yu, “Content-Centric Sparse Multicast Beamforming for Cache-Enabled Cloud RAN,” *Wireless Communications, IEEE Transactions on*, vol.PP, no.99, pp.1-1, 2016.

[3] A. Liu and V. K. N. Lau, “Asymptotic Scaling Laws of Wireless Ad Hoc Network With Physical Layer Caching,” *Wireless Communications, IEEE Transactions on*, vol. 15, no. 3, pp. 1657-1664, March 2016.

[4] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp.388-404, Mar 2000.

[5] A. Ozgur, O. Leveque and D. Tse, “Hierarchical Cooperation Achieves Optimal Capacity Scaling in Ad Hoc Networks,” *Information Theory, IEEE Transactions on*, vol. 53, no. 10, pp. 3549-3572, Oct. 2007.

[6] M. Ji, G. Caire and A. F. Molisch, “Optimal throughput-outage trade-off in wireless one-hop caching networks,” *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pp.1461-1465, July 2013.

[7] M. Ji, G. Caire and A. F. Molisch, “Fundamental Limits of Caching in Wireless D2D Networks,” *Information Theory, IEEE Transactions on*, vol.62, no.2, pp.849-869, Feb. 2016.

[8] S. W. Jeon, S. N. Hong, M. Ji and G. Caire, “Caching in wireless multihop device-to-device networks,” *IEEE International Conference on Communications (ICC)*, pp. 6732-6737, 2015.

[9] S. W. Jeon, S. N. Hong, M. Ji, G. Caire and A. F. Molisch, “Wireless Multihop Device-to-Device Caching Networks,” *Information Theory, IEEE Transactions on*, vol. 63, no. 3, pp. 1662-1676, March 2017.

[10] A. Ozgur, O. Leveque and D. Tse, “Spatial Degrees of Freedom of Large Distributed MIMO Systems and Wireless Ad Hoc Networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 2, pp. 202-214, February 2013.

[11] M. A. Maddah-Ali and U. Niesen, “Fundamental Limits of Caching,” *Information Theory, IEEE Transactions on*, vol.60, no.5, pp.2856-2867,

May 201

4.

[12] A. Ozgur and O. Leveque, “Throughput-delay trade-off for hierarchical cooperation in ad hoc wireless networks,” *2008 International Conference on Telecommunications, St. Petersburg*, 2008, pp. 1-5.

$$\square = \square + \square \quad \square , \text{ and } \square = 1-(\square - \square) . \qquad (28)$$

Applying L’Hospital’s Rule, we have that

$$\lim_{\square \to \infty} \frac{\square}{\square^{\square} \square^{-} \square} \overset{(\square)}{=} \lim_{\square \to \infty} \frac{\square \square^{\square - \square + 1}}{\square \square^{\square^{\square}}} = 0, \qquad (29)$$

where the equality (a) is obtained by keeping differentiating the numerator and the denominator until the power exponent of the numerator is negative. Thus, we have

$$\lim_{\square \to \infty} \square_{\square \square \square} = 0. \qquad (30)$$

meaning that the scaling $\square_{\square} = \Theta\left(\frac{\square}{\square^{\square+1}}\right)$ is achieved with high probability.

This finishes the proof of Theorem 1.

[13] S. N. Hong and G. Caire, “Beyond Scaling Laws: On the Rate Performance of Dense Device-to-Device Wireless Networks” *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 4735-4750, Sept. 2015.

[14] J. Guo, J. Yuan and J. Zhang “The Throughput Scaling Law of Wireless Device-to-device Caching Networks with Distributed MIMO and Hierarchical Cooperations,” *submitted to Wireless Communications, IEEE Transactions on*, 2017.