Scaling Laws for Cache-Aided Device-to-Device Networks for Wireless Video

Ming-Chun Lee, Mingyue Ji, and Andreas F. Molisch

June 2 2020

1 Introduction

1.1 Motivation for D2D-based video distribution

Wireless data traffic has been increasing by $50\% - 100\%$ per year over the past decade, and is expected to continue this spectacular growth. The majority of this video traffic (currently $> 65\%$ and growing) stems from wireless video, in particular video streaming, i.e., services like Netflix, Disney+, and Youtube. Essentially, video streaming has emerged as the "killer application" of 4G and early 5G wireless systems.

At the same time, the large and increasing data traffic strains the resources of the networks. Traditional methods for increasing network capacity include addition of new spectrum, network densification, heterogeneous networks, and use of massive antenna arrays. However, all of these solutions are expensive as they require either license fees or new infrastructure. Furthermore, they are *not scalable* - both spectrum, and locations at which BSs (and their associated backhaul) can be placed are limited.

Yet, wireless video has a special property that allows a much more efficient distribution than most other types of data files: it has a very concentrated popularity distribution, i.e., a relatively small number of popular movies/clips accounts for the majority of the data traffic. In traditional TV, this fact was exploited by the broadcast effect, such that the system throughput per user was independent of the number of users. Of course, this was achieved by forcing users to watch a pre-determined video at a pre-determined time. *Asynchronous content reuse*, which normally requires transmission at unpredictable times, to each user separately, can be made more efficient by caching of content at the devices; the downloading to the devices can then occur at times when it does not affect overall network performance, e.g., during the night. In its simplest form, called self-caching, a device would cache the videos that they anticipate its owner to watch (this is implemented, e.g., in Netflix's"smart download"). However, due to the limited storage space on cellphones and similar devices, the percentage of files that can be downloaded is small and does not lead to a significant relief of the network.

A solution for this problem was first suggested in [18–20]: different devices, which are in close proximity, and which have different videos cached, exchange (upon request) files through a spectrally efficient device-to-device communications process. If at the time of caching it is known which devices will be close together during the file exchange, a deterministic choice of which device caches which video can be made; otherwise, each device can determine, independently and randomly (but according to a given probability density function), which videos it will cache. A group of close-together devices can then be seen as "pooling" their caching resources, while efficiently communicating from the device that has the video stored, to the device that requests it. The more caching space is available on each device, the less bandwidth is required for communication (desired files are closer to the requesting user, so that area spectral efficiency of the communication is increased). This thus allows to "turn memory into bandwidth". Considering that memory is a relatively cheap resource, and bandwidth is the most expensive one in wireless systems, this is a very attractive tradeoff. Simulations under realistic settings indicated that network throughput can be increased by one to two orders of magnitude. Consequently, this topic has been investigated extensively over the past 9 years, and some 1000 papers have been published - various aspects are discussed in other parts of this book, and the reviews [5, 41, 48].

A fundamental question is whether the approach is scalable: in particular, as the density of users increases, are the benefits of caching retained, or do they peter out? From intuition, one might hope that caching benefits are retained: the number of file requesters, and the number of available caches, both increase proportionately as the user density increases. Yet, other effects, such as the increase of the file library (i.e., the collection of videos that the users might want to see) might create problems. Scaling laws aim to obtain the fundamental behavior of throughput and other relevant quantities as user density becomes very large. By providing lower and upper bounds (achievable bound and converse), we can judge not only the potential of a scheme, but also whether an implementable scheme might be close to the theoretical optimum.

In this chapter, we will first review scaling laws for "standard" (non-caching) D2D networks, as well as other caching approaches that do not make use of D2D communications (see Sec. ?.1.b). Sec. 2 the summarizes the system model that underlies most scaling law investigations. This is followed by a review of scaling laws for single-hop and multi-hop networks in Secs. 3 and 4, respectively. Next, we discuss the scaling laws when coded caching is combined with D2D communications in Sec. 5. A discussion of how the scaling laws relate to practical implementation rounds off the chapter.

1.2 D2D scaling laws without caching

Cache-aided D2D networks show considerable similarity to ad-hoc networks. In both cases, communications occur not through a central infrastructure node, but directly between devices. The main difference is that in ad-hoc networks a device might want to talk to a specific other device, while in cache-aided D2D

networks, a device wants to receive a *content*, and does not care which device sends it. Still, scaling laws for ad-hoc networks are an important reference point and many of the techniques developed for them can be adapted for cache-aided D2D networks.

A milestone in scaling laws for ad-hoc networks was the work by Gupta and Kumar [22], which investigated the transport capacity when multi-hop relaying was allowed. Placing N users that are either placed arbitrarily or randomly, both a lower (achievable) bound on the throughput per user, of the order $\Theta\left(\frac{1}{\sqrt{N \log N}}\right)$ and an upper bound (under some conditions) of $\Theta\left(\frac{1}{\sqrt{N}}\right)$ were derived. In [4], a similar analysis was conducted with a more general physical model and the upper bound $\Theta\left(\frac{1}{\sqrt{N}}\right)$ was validated under general conditions. The $\Theta(\sqrt{\log N})$ gap between the achievable throughput and the upper bound was closed in [13], however, with a slightly different model where the user distribution is described by a Poisson point process (PPP). A number of other schemes and channel models were investigated in other papers as well.

While all the above techniques were based on single-hop or multi-hop relaying, a more complicated transmission scheme can actually increase the throughput in a scaling law sense. Ref. [56] introduced a hierarchical cooperation scheme, where the cooperation between users is employed to form a distributed multiple-input multiple-output (MIMO) system among the users. The resulting scaling of the throughput per user is almost $\Theta(1)$, at the price of very complicated cooperation among user nodes. Finally, besides the scaling laws for throughput also those for the throughput-delay tradeoff have been investigated [11, 12, 14].

Other scaling laws were developed in the computer science literature, with content delivery networks in mind. Assuming *wired* connections, [10] proposed a caching policy (square-root replication policy) that provides the optimum design in terms of the *expected number of nodes* to visit until finding the desired content.

An important alternative to D2D-based caching networks is the so-called *coded caching*, in which devices locally cache part of the files. Then, the base station broadcasts a network-coded transmission such that the each device can recover the desired files from the stored parts of the files and the broadcast transmission; thus all devices can recover the desired files simultaneously. The seminal paper of [46] provided the scaling laws which are - as will be discussed more in Sec. 5, comparable to those achievable with cache-aided D2D networks, though the latter perform better in realistic circumstances.

Of course, the approach that is used in currently implemented networks, namely treating each file transmission as a separate unicast transmission, has a scaling law for the throughput as well - the throughput per user decreases as $\Theta(1/N)$.

2 System model

2.1 Popularity model

The scaling laws discussed in this chapter assume a certain popularity distribution of the files. More precisely, it is assumed that user $u \in \mathcal{U}$ requests a file $\mathsf{f}_u \in \mathcal{F} = \{1, \ldots, M\}$ in an i.i.d. manner, according to a given request probability mass function $\{P_r(f) : f \in \mathcal{F}\}$. Here $\mathcal{F} = \{1, \ldots, M\}$ denotes the "file library", from which the requests are made. Note that this library contains only the files the users might be interested in, and thus M may change as the number of users changes (see below). We note that the assumption of i.i.d. requests might be violated in practice, either because different users have different tastes and thus different request probability, and/or because different users influence each other, e.g., through social networks. Yet, the i.i.d. model is used in essentially all scaling law investigations, because it enables much better mathematical tractability. Moreover, it is assumed that different users making the requests on the same file would request different segments of the file (e.g., due to different start times for streaming a video), which means that naive multicasting does not provide any gain. It is furthermore commonly assumed that all files have the same size; if large files exist, they can be broken into smaller chunks.

The most common assumption for the probability mass function (PMF) of the file popularity is that P_r is a Zipf distribution with parameter γ [9], i.e.,

$$P_r(f) = \frac{f^{-\gamma}}{\sum_{j=1}^{M} j^{-\gamma}} \tag{1}$$

for $f = 1, \ldots, M$. This distribution is not only well-suited for closed-form analysis, but also was shown to be in agreement with early investigations into the popularity of Youtube videos [9]. The Zipf parameter γ can be interpreted as a measure for the popularity concentration - the larger γ, the more a few popular videos determine traffic. It also has important mathematical consequences: for $\gamma < 1$, the distribution is heavy-tailed, and for $M \to \infty$, $P_r(f) \to 0$ for all f. For $\gamma > 1$, the sum in the denominator of (1) sums to a finite value for $M \to \infty$, i.e., the distribution has a fast-decaying tail.

A more general distribution is the MZipf distribution [24] :

$$P_r(f) = \frac{(f+q)^{-\gamma}}{\sum_{j=1}^{M}(j+q)^{-\gamma}}, f = 1, 2, ..., M. \tag{2}$$

The parameter q denotes the "plateau" factor, such that the q most popular files have almost the same popularity, while beyond that the PMF decays, essentially, like a Zipf distribution. Ref. [40] showed that for an extensive real-world dataset of long-form videos (TV shows and movies), the MZipf distribution better fits the requesting behaviors of mobile users. In practice, the dataset in [40] showed that $q \ll M$.

2.2 Network model

The scaling analysis generally considers a network deployed over a unit-area squared region and formed by N nodes $\mathcal{U} = \{1, \ldots, N\}$. Some papers place those nodes on a regular grid with minimum node distance $1/\sqrt{N}$, while others assume that the nodes are distributed according to either a binomial Point Process (BPP) or a homogeneous Poisson Point Process (PPP); Secs. 3-6 will specify which results are derived under which of those assumptions. Each node has a cache of size S (in units of files).

A scaling analysis can generally consider two approaches for the caching of files on the nodes: deterministic and stochastic. In the deterministic approach, it is assumed that during the distribution of the files to the caches, the location of all devices and the file transmission scheme during the demand phase will be known; the caching content for each node can then be deterministically optimized. In a "clustering scheme" (see below for details), this can be done, e.g., such that each file is cached exactly once in each cluster. In the most common random approach, namely a simple "decentralized" random caching strategy, each user caches S files selected independently from the library $\mathcal{F}$ with probability $P_c(f)$, where $0 \leq P_c(f) \leq 1$ and $\sum_{f=1}^{M} P_c(f) = 1$. The caching probability distribution, $P_c(f)$, which is by definition common to all users, can be optimized by the network designer. Note that when using this policy, a user might cache the same file multiple times, and this policy is used for the sake of analysis. To avoid caching the same file multiple times, an improved random caching strategy that considers $\sum_{f=1}^{M} P_c(f) = S$ and implemented via the approach proposed in [8] can be adopted.

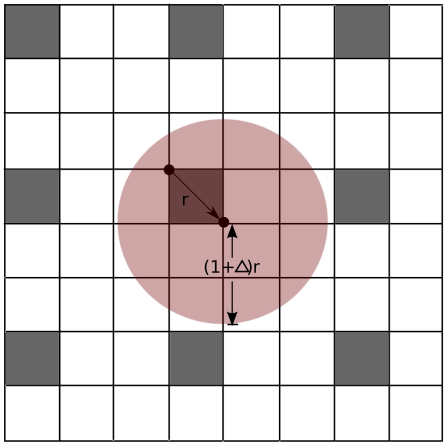


Figure 1: Single-cell layout and the interference avoidance TDMA scheme. Each square represents a cluster, with grey squares representing concurrently transmitting clusters. The red area is the disk where the protocol model allows no other concurrent transmission.

For the delivery of the files to the requesting user, the cluster model is the most common model for the achievability bounds. The network is divided into clusters of equal size, such that the number of nodes in each cluster is $g_c(M)$, which are independent of the users' requests and cache placement realization. A user can look for the requested file only inside its own cluster. If a user can find the requested file inside the cluster, we say there is one *potential link* in this cluster. An *interference avoidance* scheme ensures that at most one

transmission is allowed in each cluster on any time-frequency slot (transmission resource), while inter-cluster interference is avoided by a time-frequency reuse scheme [50, Ch. 17] with parameter K as shown in Fig. 2.2. In particular, we can pick $K = \left(\lceil \sqrt{2}(1+\Delta) \rceil + 1\right)^2$, where Δ is the interference parameter in the protocol model.

A system admission control scheme decides whether to serve potential links or ignore them. The served potential links in the same cluster are scheduled with equal probability (or, equivalently, in round robin), such that all admitted user requests have the same average throughput $\mathbb{E}[T_u] = \overline{T}_{\min}$ (see Section 2.3), for all users u, where expectation is with respect to the random user requests, random caching, and the link scheduling policy (which may be randomized or deterministic, as a special case).

2.3 Target functions for scaling

Ultimately, the goal of the scaling analysis is to provide the asymptotic behavior of the target function when $N \to \infty$, $M \to \infty$. The simplest target function is the sum throughput (or equivalently, the average per-user throughput), which was used in some of the earlier scaling analyses. However, this quantity has the drawback that it can take on very large values while leading to fundamental unfairness in the system - essentially, it might induce the caching distribution to concentrate on the most popular files and thus boost throughput, but neglecting to serve users with other requests.

For this reason, later papers concentrated on the throughput-outage tradeoff. Qualitatively (for formal definition see [28]), we say that a user is in outage if the user cannot be served by the D2D network. This can be caused by: (i) the file requested by the user is not in the user's own cluster, (ii) that the system admission control decides to ignore the request. We define the outage probability p_o as the average fraction of users in outage.

More precisely, it is common to define for a given network and request probability mass function, an outage-throughput pair (p, t) as *achievable* if there exists a cache placement scheme and an admission control and transmission scheduling policy with outage probability $p_o \leq p$ and minimum per-user average throughput $\overline{T}_{\min} \geq t$. The outage-throughput achievable region $\mathcal{T}(P_r, n, m)$ is the closure of all achievable outage-throughput pairs (p, t). In particular, we let $T^*(p) = \sup\{t : (p, t) \in \mathcal{T}(P_r, n, m)\}$. $\Diamond$

Notice that $T^*(p)$ is the result of the optimization problem:

$$\begin{aligned} \text{maximize} \quad & \overline{T}_{\min} \\ \text{subject to} \quad & p_o \leq p, \end{aligned} \tag{3}$$

where the maximization is with respect to the cache placement and transmission policies. Since the scaling law analysis is essentially to characterize the asymptotic behavior of the networks, the scaling law order notations defined in Table 1 will be intensively used below.

Table 1: Scaling Law Order Notations

Scaling law notations	Mathematical definitions	Asymptotic interpretations
$f(n) = \mathcal{O}(g(n))$	$\exists$ a constant c and integer N such that $f(n) \leq cg(n)$ for $n > N$	$f(n) \leq g(n)$
$f(n) = o(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$	$f(n) \ll g(n)$
$f(n) = \Omega(g(n))$	$g(n) = \mathcal{O}(f(n))$	$f(n) \geq g(n)$
$f(n) = \omega(g(n))$	$g(n) = o(f(n))$	$f(n) \gg g(n)$
$f(n) = \Theta(g(n))$	$f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(f(n))$	$f(n) = g(n)$

3 Single-hop networks

In this section, we discuss scaling law results of cache-aided D2D networks considering uncoded single-hop communications.[1] The main benefit of considering such networks is that the implementation can remain simple when files are delivered. Additionally, single-hop communications are more plausible in current cellular systems [42, 54]. We will in the following first discuss the theoretical studies, and then provide numerical results under realistic setups to support the theories.

3.1 Scaling law results

The first work investigating the scaling law of cache-aided single-hop uncoded D2D networks was [17]. This paper investigated the scaling law of the maximum expected throughput, under the assumptions that the popularity model is characterized by the Zipf distribution, the protocol model is adopted for communications, and N users are distributed within the network uniformly at random. The results indicate that the throughput per user can scale with $\Theta(1)$ when $\gamma > 1$ and scale with $\Theta\left(\frac{1}{M^\eta}\right)$, where $\eta = \frac{1-\gamma}{2-\gamma}$, when $\gamma < 1$.[2] These scaling laws are achieved via adopting a caching policy that caches files according to a Zipf distribution, and then optimizes the collaboration distance in order to pursue the maximum expected number of links in the network. These scaling law results indicate that the expected number of links, and thus the throughput, that can be offloaded from the BSs to the D2D network is significant as the expected throughput scales linearly with N when $\gamma > 1$ and scales with $\Theta\left(\frac{N}{M^\eta}\right)$ when $\gamma < 1$.

Although results in [17] showed promising offloading performance when using cache-aided D2D networks, the outage probability was ignored. In fact, it was found later that when $\gamma \in (0, 1)$ (heavy-tail popularity distribution), this maximum expected throughput can only be achieved when the outage probability goes to 1 as $N \to \infty$. As a consequence, only a small portion of the users are served, while the majority of the users are left without having services (i.e., in

[1] "Uncoded" here means that no inter-file coded schemes will be used.

[2] To be more specific, the $\Theta\left(\frac{1}{M^\eta}\right)$ scaling law when $\gamma < 1$ is a performance outer bound, and there is a small gap between the achievable performance and this value.

outage). Ref. [28] discovered this problem and investigated the scaling behavior in terms of throughput-outage tradeoff. It provides a tight characterization for this tradeoff in the order sense.

Ref. [28] assumes a grid network for the user distribution and a protocol model for the communication links. The discussions in [28] focused on results under the assumption that the popularity distribution is Zipf distribution. The model was later generalized in [40] to the MZipf distribution, as it is of practical interest how the plateau factor q of a MZipf distribution influences the performance. For q is a constant while $M \to \infty$, the MZipf distribution would behave identical to a Zipf distribution, while consideration of $q = \mathcal{O}(M) \to \infty$ reveals the influence of the plateau factor q. In addition, the MZipf distribution can converge to simple uniform distribution when $q = \omega(M)$.

To obtain the lower performance bound, Ref. [28] and [40] consider the following achievable scheme. The transmission policy is based on *clustering* described in Sec. 2, and the caching policy on *independent random caching*. Only a single link can be activated at a time in the cluster because single-hop D2D is considered. The independent random caching policy that minimizes the outage probability in a cluster is as follows (Theorem 1 in [40]):

Theorem: Define $c_2 = qa'$, where $a' = \frac{\gamma}{S(g_c(M)-1)-1}$. Let $c_1 \geq 1$ be the solution of the equality $c_1 = 1 + c_2 \log\left(1 + \frac{c_1}{c_2}\right)$. Let $M \to \infty$ and $N \to \infty$. Suppose $g_c(M) \to \infty$ as $M \to \infty$, and denote m^* as the smallest index such that $P_c^*(m^* + 1) = 0$. The the caching distribution $P_c^*(\cdot)$ that maximizes the probability that any user u finds its requested file inside the corresponding cluster is:

$$P_c^*(f) = \left[1 - \frac{\nu}{z_f}\right]^+, f = 1, ..., M, \tag{4}$$

where $\nu = \frac{m^*-1}{\sum_{f=1}^{m^*} \frac{1}{z_f}}$, $z_f = (P_r(f))^{\frac{1}{S(g_c(M)-1)-1}}$, $[x]^+ = \max(x, 0)$, and

$$m^* = \Theta\left(\min\left(\frac{c_1 S g_c(M)}{\gamma}, M\right)\right). \tag{5}$$

□

From the theorem, we observe that $P_c^*(f)$ is a monotonically decreasing function, indicating that the optimal policy generally is to cache the file requested more times with high probability. Besides, when $q = \Theta(1)$ is a constant, $c_1 \to 1$ can be observed, and the theorem applies to networks that consider Zipf distributions (Theorem 4 in [28]). The influence of q on the optimal caching policy is realized through c_1 and c_2. We observe that $c_1 = \mathcal{O}(c_2)$ when $c_2 = \Omega(1)$. Thus, when considering $q = \Omega\left(\frac{S g_c(M)}{\gamma}\right)$ and $\frac{c_1 S g_c(M)}{\gamma} < M$, we obtain $m^* = \Theta(\frac{c_1 S g_c(M)}{\gamma}) = \Theta\left(\sqrt{\frac{S q g_c(M)}{\gamma}}\right)$. Combining above results, this theorem indicates that the number of files that the caching policy should cover is relevant not only to the number of nodes in a cluster but also to the rank q (order-wise) in the library. This is intuitive because the MZipf distribution

has a relatively flat plateau regime and q characterizes the break point between the plateau and rolloff regimes. The following expositions assume that $P_c(f) = P_c^*(f)$.

The practically most interesting cases occur for the regime with small outage probability. Considering first the situation with $\gamma < 1$, it is found

Theorem (Theorem 2 in [40]): Let $M \to \infty$ and $N \to \infty$. Suppose $g_c(M) \to \infty$ as $M \to \infty$. Consider $M = \mathcal{O}(N)$, $SN \gg M$, $q = \mathcal{O}\left(\frac{Sg_c(M)}{\gamma}\right)$, and $\gamma < 1$. Define $D = \frac{q}{M}$. When $g_c(M) = \frac{\rho M}{c_1 S}$, where $\rho \geq \gamma$, the achievable throughput-outage tradeoff is:

$$\begin{aligned} T(P) &= \frac{C}{K}\frac{Sc_1}{\rho M}, \\ p &= \frac{(1-\gamma)e^{-(\rho/c_1-\gamma)}}{(1+D)^{1-\gamma}-(D)^{1-\gamma}} \\ &\quad \cdot \left[(1+D)^{\frac{\gamma}{S(g_c(M)-1)-1}+1} - (D)^{\frac{\gamma}{S(g_c(M)-1)-1}+1}\right]^{-(S(g_c(M)-1)-1)}. \end{aligned} \tag{6}$$

□

By choosing $g_c(M) = \beta M$ for some $\beta > 0$, it is apparent from the theorem that p strictly bounded away from 1. By fixing a small but positive target outage probability, the per-user average throughput of the D2D one-hop caching network with random (decentralized) caching scales as $T(p) = \Theta\left(\max\left\{\frac{1}{N}, \frac{S}{M}\right\}\right)$, where the scaling $\Theta\left(\frac{1}{N}\right)$ can be trivially achieved by letting the whole network to be a single cluster and serving one demand per unit time. This scaling is equivalent to conventional unicasting from a single omniscient node which can be regarded as the state of the art of today's (single cell) systems, with a base station or access point serving individual requests without exploiting the asynchronous content reuse. We notice that when $SN \gg M$, the throughput of the D2D caching network achieves per-user throughput that increases linearly with S. This indicates that caching in the user nodes and exploiting the spatial reuse of the D2D network is a very attractive approach in dense networks, since storage space is much "cheaper" than scarce resources such as bandwidth or dense base station deployment. Since the benefits of uncoded caching is significant under the assumption that $SN \gg M$, the reminder of Secs. 3 and 4 considers such an assumption unless otherwise stated. Finally, we note that when considering a Zipf (i.e., $q = \Theta(1)$), we have $p = (1-\gamma)e^{\gamma-\rho}$, corresponding to the results in Theorem 5 of [28].

Apart from the achievable scheme, Ref. [28] also showed that the scaling law outer bound of the considered network is identical to the achievable performance discussed above, i.e., $T(p) = \Theta\left(\frac{S}{M}\right)$ when the outage probability is negligible. This validates the optimality of the scaling law performance here. Note that although the outer bound in [28] was derived considering of Zipf distribution, it is also valid for the MZipf distribution, as increasing q can only degrade the performance for a given random caching policy (see Appendix D in [40]).

To understand the potential of cache-aided D2D networks, it is instructive to compare the scaling laws achieved by the D2D caching network with those

achievable by other possible approaches. We have already discussed conventional unicast, achieving $\Theta(\frac{1}{N})$ throughput. When the number of files M is less than the number of users N, an alternative and well-known approach is Harmonic Broadcasting [34]. In brief, Harmonic Broadcasting works as follows: fix the maximum waiting delay of τ "chunks" (from the time a streaming session is initiated to the time playback is started), and let L denote the total length of the video file, expressed in chunks. In Harmonic Broadcasting, the video file is split into successive blocks such that for $i = 1, \ldots, \lceil L/\tau \rceil$, there are i blocks of length τ/i. Then, each i-th set of blocks of length τ/i is repeated periodically on a (logical) subchannel of throughput R/i, where R is the transmission rate (in bit/s) of the video playback. Users receive these channels in parallel. In this way, each file requires a downlink rate of $R\log(L/\tau)$. Hence, the total number of files that can be sent in the common downlink stream is $m' = \min\left\{\frac{C_{r_0}}{R\log(L/\tau)}, M\right\}$, yielding an average throughput per user of $R\,(1-p_o)$ with outage probability $p_o = \sum_{f=m'+1}^{M} P_r(f)$, since all requests to files not included in the common downlink stream are necessarily in outage. Accordingly, throughput per user of Harmonic Broadcasting scales as $\Theta\left(\frac{1}{m'\log\frac{L}{\tau}}\left(1-\sum_{f=m'+1}^{M} P_r(f)\right)\right)$, where $m' \leq M$. It follows that when $\gamma < 1$, the throughput of Harmonic Broadcasting scales as $\Theta(\frac{1}{M\log L})$ for an outage probability that is bounded away from 1. From a strictly technical viewpoint, since in the system assumptions we study the system performance by first letting $L \to \infty$, and then considering N, M that simultaneously grow in some relation, the throughput of Harmonic Broadcasting under our assumption is identically zero. In practice, for large but finite L, the gain of D2D caching over Harmonic Broadcasting can be appreciated by comparing the term $\frac{S}{M}$ with the term $\frac{1}{M\log L}$ in the per-user throughput. It is clear that Harmonic Broadcasting does not take advantage of the user nodes storage memory, and in addition suffers an arbitrarily large multiplicative penalty as the length of the files L increases.

It is interesting to compare cache-aided D2D with the coded multicasting (coded-caching) scheme in [46], which represents another example of one-hop network with caching in the user nodes, able to make efficient (and in fact, near-optimal in an information theoretic sense) use of caching. The rate analysis provided in [46] shows that for large M, N and finite S, the scaling of the per-user throughput of the coded multicasting is $\Theta\left(\max\left\{\frac{1}{N}, \frac{S}{M}\right\}\right)$, where the two terms inside the max are realized depending on whether $NS \gg M$ or not. Interestingly and somehow surprisingly, this is the same scaling behavior of the D2D caching network derived above. However, coded multicasting requires higher implementation complexity due to the need for network coding and is vulnerable to fading effects, as shown in the simulations below.

The above results are based on the assumption that $\gamma < 1$. The following theorem characterizes the scaling law under the assumption that $\gamma > 1$.

Theorem (Theorem 4 in [40]): Let $M \to \infty$, $N \to \infty$, and $q \to \infty$. Suppose $g_c(M) \to \infty$ as $M \to \infty$. Consider $\gamma > 1$, $g_c(M) = o(M) \leq N$, and $q =$

$\mathcal{O}\left(\frac{Sg_c(M)}{\gamma}\right)$. Define $c_6 = \frac{q}{g_c(M)}$. The achievable throughput–outage tradeoff is

$$T(p) = \frac{C}{K}\frac{1}{g_c(M)} + o\left(\frac{1}{g_c(M)}\right), \quad p = (c_6)^{\gamma-1}\frac{Sc_1 + c_6}{\left(\frac{Sc_1}{\gamma} + c_6\right)^{\gamma}}. \tag{7}$$

□

It should be noted that this theorem considers $g_c(M) = o(M)$ and $q = \mathcal{O}(g_c(M))$. This is because such a consideration leads to a more interesting scaling law, as it breaks the $T(p) = \Theta\left(\frac{S}{M}\right)$ limit as compared to the case that $\gamma < 1$. As a matter of practice, we observe that $q \ll M$ from the dataset in [40]. From the theorem, we observe that if we let $g_c(M) = \Theta(q)$, the achievable scaling law is $\Theta(\frac{S}{q})$ when $NS \gg q$. This implies that when $\gamma > 1$ and the aggregate memory is larger than the order of q, the improvement of the cache-aided D2D is significant even if we have a large M. This relaxes the condition that applies for $\gamma < 1$, namely that a small library (i.e., $NS \gg M$) is required to have significant benefits. Furthermore, the fact that the real dataset in [40] has $q \ll M$ and $\gamma > 1$ for mobile users strongly supports the practical significance of such a result. Finally, it should be noted that the case considering a Zipf distribution with $\gamma > 1$ can be viewed as either a special case of the MZipf distribution with $q = \mathcal{O}(1)$ and $\gamma > 1$ or a special case of the Zipf distribution with $M = \mathcal{O}(1)$ and $\gamma < 1$. The latter one holds because when considering a Zipf distribution with $\gamma > 1$, a finite number of files collects essentially all the request probability mass. Accordingly, we can conclude that the scaling law with $\Theta\left(\frac{S}{M^{\epsilon}}\right)$, where ϵ is arbitrarily small, is achievable with negligibly small outage probability when a Zipf distribution with $\gamma > 1$ is considered.

3.2 Numerical studies under realistic setups

Refs. [30] and [40] provided abundant simulations under realistic setups to show the advantages of cache-aided D2D networks. As an example, we discuss here the "mixed-deployment" scenario described in [30]. A 0.36km^2 (600m × 600m) area that contains buildings as well as streets/outdoor environments is considered. $N = 10000$ users are uniformly and independently distributed in the cell, i.e., on average, there are $2 \sim 3$ nodes in each square of 10×10 meters. The cell contains a Manhattan grid of square buildings with side length of 50m, separated by streets of width 10m. Each building is made up of offices of size 6.2m × 6.2m. Within the cell, users (devices) are distributed at random according to a uniform distribution. Each node is assigned to be outdoors or indoors, and in the latter case in a particular office, according to where its location falls on the map of the environment. D2D communications in the 2.4 GHz band is considered. Depending on the locations of the two users in the D2D communication pair, the employed channel model is for indoor communication (Winner model A1), outdoor-to-indoor communication (B4), indoor-to-outdoor communication (A2), and outdoor communication (B1).

In particular, the simulations directly use the appropriate Winner II channel models with antenna heights of 1.5m, and the corresponding probabilistic Line of Sight (LOS) and Non Line of Sight (NLOS) models. A probabilistic body shadowing loss (σ_{L_b}) with a lognormal distribution accounts for the blockage of radiation by the person holding the device, where for LOS, $\sigma_{L_b} = 4.2$ and for NLOS, $\sigma_{L_b} = 3.6$ [36].

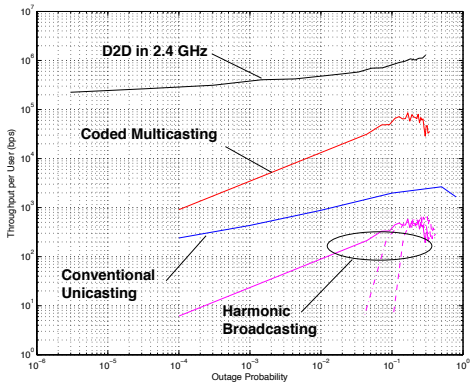


Figure 2: Simulation results for the throughput-outage tradeoff for conventional unicasting, coded multicasting, harmonic broadcasting and the 2.4 GHz D2D communication scheme under indoor office channel models. For harmonic broadcasting with only the m' most popular files, solid line: $m' = 300$; dash-dot line: $m' = 280$; dash line: $m' = 250$. We have $N = 10000$, $M = 300$, $S = 20$ and $\gamma = 0.4$.

Fig. 2 compares throughput/outage for cache-aided D2D, conventional unicasting, harmonic broadcasting, and coded multicasting. The number of files in the library is $M = 300$. The user cache size is $S = 20$ files, which even with high definition (HD) quality requires less than the (nowadays) ubiquitous 64 GByte of storage space. Each user independently makes a request by sampling from a Zipf distribution with $\gamma = 0.4$; this value is at the lower edge of the range of values that have been measured in practice [9]; note that the advantages of caching would be *more* pronounced for larger γ. The interference between concurrent D2D links sharing the same frequency band is treated as noise. The harmonic broadcasting uses a video file size of $L = 5400$ chunks and $\tau = 10$ chunks, so that the number of blocks is $L_N = \lceil \frac{L}{\tau} \rceil = 540$ [59].

We can see that the throughput of the D2D scheme is markedly (orders of magnitude at low outages) higher than the conventional unicasting, harmonic broadcasting, and even coded multicast. This shows that in practical situations, the "scaling law" is not the only aspect of importance. Rather, the higher capacity of the short-distance links plays a significant role, and a good throughput-outage tradeoff can be achieved even without the use of a BS connection as "backstop". The main reason lies in the fact that for the coded multicasting or harmonic broadcasting scheme, outage is determined by bad channel conditions, and no diversity is built into the system. For D2D the channel diversity plays a more importance role.

To understand the impact of a different popularity distribution, Fig. 3 shows results in similar settings, but for an MZipf distribution. Specifically, the MZipf distribution with $\gamma = 1.16$, $q = 22$, and $M = 7345$ is considered, where these parameters are extracted from the July Metro region 1 dataset defined in [40]. Fig. 3 shows the throughput-outage tradeoff for different cache sizes on each device. We observe that the throughput of 10^5 bps can be achieved if the cache size of each user is up to 1/10 of the library size, showing significant improvement as compared to the conventional unicasting. Even for $S = M/50$, i.e., approximate 100 files, the advantage compared to the conventional unicasting is two orders of magnitude. Even just caching of 30 files ($M/200$) also provides significant throughput gains, though only for outage probabilities > 0.01.

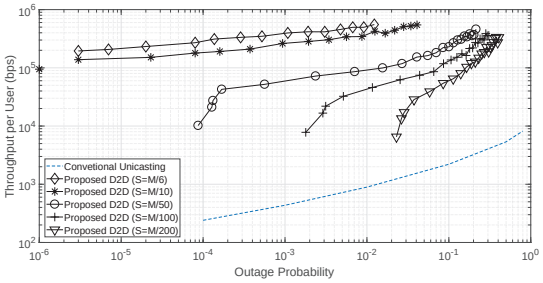


Figure 3: Throughput outage tradeoff in networks assuming office scenario for propagation channel; varying local storage size; and $\gamma = 1.16$, $q = 22$, and $M = 7345$.

Combining results in Figs. 2 and 3, we can conclude that cache-aided D2D networks show significant improvement, as compared to other video distribution schemes. Besides, by comparing Figs. 2 and 3, we observe that even though the scaling law changes when changing the popularity distribution from one to another, the performance of the cache-aided D2D network is not very sensitive to such a change, as long as the total cache space in the network is sufficient. This again indicates that the "scaling law" is not the only aspect of importance in practical systems.

4 Multi-hop networks

Although the implementation is more difficult than single-hop D2D, adopting multi-hop delivery in cache-aided D2D networks can significantly improve scaling laws. We thus in this section discuss scaling laws of cache-aided multi-hop D2D networks.

Two multi-hop file delivery schemes have been used in the literature for scaling law analysis. The first scheme, proposed in [27], uses local multi-hop delivery with clustering. Specifically, the network is first split into clusters where interference between clusters are avoided by frequency reuse, just as in the schemes described in Sec. 3. Then, within each cluster, users obtain (if possible) files from other users in the same cluster via a multi-hop delivery scheme; a user in a cluster can only access files cached by users in the same

cluster. The local multi-hop delivery scheme within each cluster operates as follows. First, the cluster is split into small equally-sized square hopping cells. Then, users requesting files are paired with other users who cache their desired files, constructing source-destination pairs. Each source-destination pair defines their vertical and horizontal hopping path that connects the source and destination. To deliver the file, the source user first transmits the desired file through the horizontal path hopping along adjacent hopping cells, and then hopping through the vertical path, as illustrated in Fig. 4. When several source-destination pairs in the cluster transmit files, a hopping cell might need to relay multiple files to its adjacent cells. In this case, a round-robin approach is used for all files in the same hopping cell. Note that to avoid interference between different hopping cells, a TDMA scheme with reuse factor J is used as the interference avoidance scheme for which a hopping cell is activated every J time-slots. The throughput-outage tradeoff is then obtained by adjusting the sizes of clusters and hopping cells, leading to the desired scaling law.

The second scheme, which was proposed in [39], also adopts the hybrid clustering and multi-hop delivery structure, while a different multi-hop delivery approach is used within each cluster. Specifically, the multi-hop delivery scheme proposed in [13] is exploited within each cluster for file delivery in this case. Denote $\mathcal{V}_f$ as the set of users in a cluster that cache file f. Then for each user u in the cluster, if the requested file f can be found in the caches of users in $\mathcal{V}_f$, then a user v_f, randomly selected from $\mathcal{V}_f$, is chosen as the source to deliver the requested (real) file f to user u. If the requested file cannot be found from the caches of any users in the cluster, user u is matched with a randomly selected user v from all users in the cluster, and then user v is set as the source for delivering a *virtual* file to user u. Note that it does not matter what file is delivered in this case, as the user is actually in outage. After the establishment of the matching of the sources and destinations, to deliver (both real and virtual) files, the multi-hop approach proposed in [13] directly applies. Such scheme is suboptimal as the delivery of virtual files cannot generate any effective throughput. Nevertheless, when the outage probability is either negligibly small or converging to zero, this scheme will be orderwise optimal because the performance degradation caused by delivering virtual files is negligible.

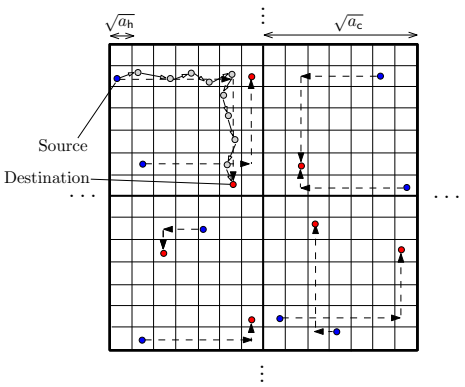


Figure 4: The multi-hop delivery after the source node selection.

Besides assuming the first of the two multi-hopping schemes, Ref. [27] assumes (i) the protocol model (ii) N users distributed according to a binomial

point process (i.e., N users uniformly and randomly distributed within the network), (iii) a heavy-tailed popularity distribution (i.e., Zipf distribution with $\gamma < 1$), (iv) $SN \gg M$, (v) a uniformly random caching policy, in which each user caches S distinct files that are selected uniformly at random from the library $\mathcal{F}$. Then the throughput scaling law $\Theta\left(\sqrt{\frac{S}{M}}N^{-\epsilon}\right)$ is achievable with negligible outage probability, where ϵ is an arbitrarily small number.

Ref. [27] also characterized the outer bound of the scaling law when the outage probability converges to zero. The key of the outer bound characterization is to obtain the minimum distances the source-destination pairs need to traverse when delivering files. Such a distance characterization is relevant to the interference regime created by a successful file delivery of a source-destination pair. Given the interference regimes of source-destination pairs, the maximum number of concurrent transmissions can be upper bounded, leading to the outer bound of the throughput. The resulting throughput scaling law outer bound is $\mathcal{O}\left(\sqrt{\frac{S}{M}}N^{\epsilon}\right)$. By comparing between the achievable performance and the outer bound, we can see that there is a $\Theta\left(N^{2\epsilon}\right) = \Theta\left(\log(N)\right)$ gap between them.

Ref. [39] adopted a somewhat more general model, in particular assuming an MZipf distributions for the file popularity, PPP for user distribution, and a physical model (interference) model for communications. It obtained achievable scaling law and outer bound for the regimes that the outage probability is either negligibly small or converging to zero and showed that they are tight. Since the Zipf distribution model is only a special case for the MZipf model, such an optimality can also be applied to networks considering Zipf distributions. This thus closes the gap mentioned above.

Specifically, when the average number of users in the network is $N \to \infty$, the scaling law analysis is conducted under the conditions that $M = o(N)$ and $q = \mathcal{O}(M)$ when $\gamma < 1$; $M = o(N)$ and $q = o(M)$ when $\gamma > 1$. The plateau factor q can either go to infinity or remain as a constant, regarded as MZipf and Zipf distributions, respectively, from the asymptotic analysis perspective. To analyze the scaling law under the above assumptions, the following caching policy minimizes the outage probability of users in a cluster:

Theorem (Theorem 1 in [39]): Let $N \to \infty$, $M \to \infty$, $q \to \infty$, and $g_c(M) \to \infty$. Denote m^* as the smallest index such that $P_c^*(m^*+1) = 0$. Let $C_2 = \frac{q\gamma}{Sg_c(M)}$; C_1 is the solution of the equation: $C_1 = 1 + C_2 \log\left(1 + \frac{C_1}{C_2}\right)$. The caching distribution $P_c^*(\cdot)$ that minimizes the outage probability p_o is as follows:

$$P_c^*(f) = \left[\log\left(\frac{z_f}{\nu}\right)\right]^+, f = 1, ..., M, \tag{8}$$

where $\nu = \exp\left(\frac{\sum_{f=1}^{m^*} \log z_f - S}{m^*}\right)$, $z_f = (P_r(f))^{\frac{1}{g_c(M)}}$, $[x]^+ = \max(x, 0)$, and

$$m^* = \Theta\left(\min\left(\frac{C_1 S g_c(M)}{\gamma}, M\right)\right). \tag{9}$$

□

This caching policy, implemented via the approach proposed in [8], has a similar structure to the one introduced in [40], while the mathematical details are different because different user distribution and communication models are used. However, the interpretation is the same – the most popular q files (orderwise) have similar request probabilities, and we need to cache them to minimize the outage probability.

When the second multi-hop file delivery scheme is considered to deliver files and the aforementioned caching policy is adopted, the network has the following achievable throughput-outage tradeoffs:

Tradeoff for $\gamma < 1$ (Corollary 1 in [39]): Let $M \to \infty$, $N \to \infty$, and $q \to \infty$. Suppose $\gamma < 1$ and $g_c(M) = \frac{\rho M}{C_1 S} = o(N)$. When considering $\rho = \Theta(1) \geq \gamma$, the following throughput-outage performance is achievable:

$$T(p) = \Omega\left(\sqrt{\frac{S}{\rho M}}\right), p = \epsilon_1(\rho), \tag{10}$$

where $\epsilon_1(\rho) > 0$ can be arbitrarily small. Furthermore, when considering $\rho \to \infty$, i.e., $\rho = \omega(1)$, we obtain the following achievable throughput-outage performance:

$$T(p) = \Omega\left(\sqrt{\frac{S}{\rho M}}\right), p = \Theta\left(e^{-\rho}\right) = o(1). \tag{11}$$

□

Tradeoff for $\gamma > 1$ (Corollary 4 in [39]): Let $M \to \infty$, $N \to \infty$, and $q \to \infty$. Suppose $\gamma > 1$ and $g_c(M) \to \infty$. Consider $g_c(M) = o(M)$, $q = o(M)$, and $g_c(M) = \frac{\alpha_1 q}{S}$. When considering $\alpha_1 = \Theta(1)$ to be large enough, the following throughput-outage performance is achievable:

$$T(p) = \Omega\left(\sqrt{\frac{S}{\alpha_1 q}}\right), p = \epsilon_2(\alpha_1), \tag{12}$$

where $\epsilon_2(\alpha_1) > 0$ can be arbitrarily small. Furthermore, when considering $\alpha_1 = \omega(1) \to \infty$, we obtain the following throughput-outage performance:

$$T(p) = \Omega\left(\sqrt{\frac{S}{\alpha_1 q}}\right), p = \Theta\left(\frac{1}{(\alpha_1)^{\gamma-1}}\right) = o(1). \tag{13}$$

□

From these results, we understand that when $\gamma < 1$ and the outage probability is negligibly small, the achievable throughput per user is $\Omega\left(\sqrt{\frac{S}{M}}\right)$. Additionally, the tradeoff between the throughput and outage probability is controlled by the parameter ρ. When considering $\gamma > 1$ and the outage probability is negligibly small, we see that the throughput per user is $\Omega\left(\sqrt{\frac{S}{q}}\right)$, and the tradeoff between throughput and outage probability is controlled by the parameter α_1. By comparing between the results of $\gamma < 1$ and $\gamma > 1$, we understand

that when the popularity distribution has a light tail ($\gamma > 1$), we can improve the scaling law as $q \ll M$ holds in practice. Finally, we note that the outer bound analysis in Theorems 4 and 5 in [39] indicate that above achievable throughput-outage performance is (orderwise) optimum as the multiplicative gap between the achievable performance and corresponding outer bound can be upper bounded by a constant. The outer bound is obtained by first characterizing the upper bound of the source-destination distances in the network when given an outage probability lower bound, and then obtain the upper bound of the throughput per user in this context. As a final remark, we note that although the achievable scheme has the assumption that a user may get a desired file from only its own cluster, the fact that this scheme can achieve (in the order sense) the outer bound shows that inclusion of inter-cluster communication does not change the scaling law.

The above achievable scheme and outer bound analysis can be conducted following the similar procedure for the case that $q = \mathcal{O}(1)$ is a constant. Such an analysis leads to scaling laws of networks considering the Zipf distribution. The results lead to the following optimal scaling laws: when $\gamma < 1$, the throughput-outage tradeoff is

$$T(p) = \Theta\left(\sqrt{\frac{S}{\rho_{\text{zip}} M}}\right), \quad p = \Theta\left(e^{-\rho_{\text{zip}}}\right), \tag{14}$$

where $\rho_{\text{zip}} = \Omega(1)$; when $\gamma > 1$, the throughput-outage tradeoff is

$$T(p) = \Theta\left(\sqrt{\frac{S}{\alpha'_{1,\text{zip}}}}\right), p = \Theta\left(\frac{1}{(\alpha'_{1,\text{zip}})^{\gamma-1}}\right), \tag{15}$$

where $\alpha'_{1,\text{zip}} = o(M)$ is any function that goes to infinity as $M \to \infty$.

Discussions in Sec. 4 to this point characterize the throughput-outage scaling laws considering decentralized caching policies. Yet, there exist some papers investigating cache-aided multi-hop D2D networks adopting centralized caching policies. In [16], the scaling law of the average throughput per node for the cache-aided multi-hop D2D network was characterized with the assumption of user locations on a grid and with a centralized caching policy. It shows that the scaling laws of $\Theta\left(\sqrt{\frac{S}{M}}\right)$ for $\gamma < 1$, of $\Theta\left(\frac{\sqrt{S}}{M^{\frac{3}{2}-\gamma}}\right)$ for $1 < \gamma < \frac{3}{2}$, and of $\Theta\left(\sqrt{S}\right)$ for $\gamma > \frac{3}{2}$ can be achieved, respectively, when a Zipf distribution is used as popularity model. Ref. [57] proposed outer bounds considering also the Zipf distribution. It showed that the outer bounds are tight to the achievable performance in [16], indicating the potential optimum scaling laws for networks adopting centralized caching policy.[3] It should be noted that when a centralized caching policy is adopted and the total cache space in the network is large enough, since the cache space can be fully controlled, the outage probability

[3]Note that [16] and [57] actually considered slightly different network models, while their insights are essentially the same.

can be exactly zero, i.e., completely no outage occurs. This is different from the decentralized random caching policy that the outage probability cannot be completely eliminated. Accordingly, the comparisons between the scaling law results of the centralized caching policy and the randomized caching policy might not be completely fair. This might also explain the additional regime that $1 < \gamma < \frac{3}{2}$ in the scaling law of the centralized caching policy, as compared to the scaling law of the randomized caching policy – the allowance of a tiny outage (either negligibly small or converging to zero) in the scaling law of the randomized caching policy eliminates such regime.

By comparing the scaling laws of networks adopting centralized policy with those adopting decentralized caching policy, we observe that they have the same scaling law when a negligibly small outage probability is allowed for the network adopting a decentralized policy. Therefore, as both centralized and decentralized caching policies have almost identical scaling laws, the use of a randomized policy is more attractive, as it is easier to implement and more robust to user mobility [49].

Finally, we would like to point out that the multi-hop schemes discussed above in this section are based on simple multi-hop transmissions without complicated physical layer processing. On the other hand, there exist papers combining caching with hierarchical cooperation [56] which enables the distributed MIMO gain during multi-hop transmissions. These papers [21, 44] can lead to better scaling laws than those introduced above at the price of very complicated cooperation among user nodes.

5 D2D Coded Caching

In this section, we will discuss the application of network coding in D2D caching networks. In particular, we will first discuss the original single-hop D2D coded caching scheme introduced in [29] and then introduce an improved scheme proposed in [68]. In the end, we will discuss the extension of D2D coded caching to multihop networks.

5.1 Single-hop Coded D2D Caching

One important constraint of the scheme described in previous sections is that both the cache placement and the delivery schemes exploit an "uncoded" approach in the delivery phase. The throughput gain is mainly obtained by effective cache placement and spatial reuse (TDMA/FDMA) if applicable. A natural question to ask is whether coded delivery (or coded multicasting) for D2D transmissions can provide an additional gain, or whether the coding gain and the spatial reuse gain can be combined. Ref. [29] (see also [31]) designs a deterministic subpacketized caching and a network-coded delivery scheme for the single-hop D2D caching networks. The scheme is explained by the example shown in Fig. 5, where we assume no spatial reuse can be used, or only one transmission per time-frequency slot is allowed but the transmission range

can cover the entire network. This scheme can be generalized to any N, M, S.

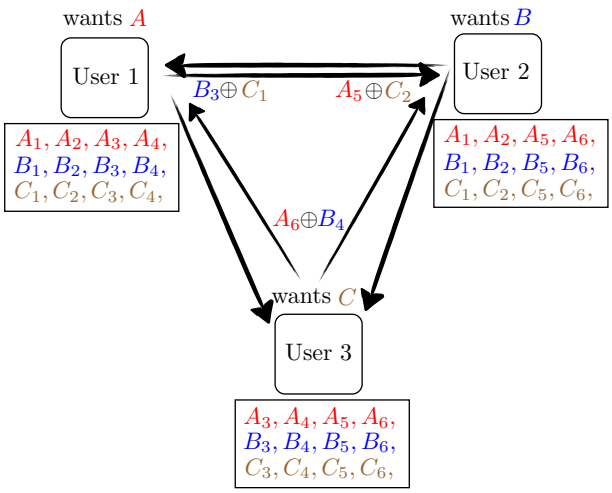


Figure 5: Illustration of the example of 3 users, 3 files and $S = 2$, achieving $1/2$ transmissions in term of file. We divide each file into 6 packets (e.g. A is divided into $A_1, \cdots, A_6$.) User 1 requests A; user 2 requests B and user 3 requests C. The cached packets are shown in the rectangles under each user. For the delivery phase, user 1 transmits $B_3 \oplus C_1$; user 2 transmits $A_5 \oplus C_2$ and user 3 transmits $A_6 \oplus B_4$. The normalized number of transmissions is $3 \cdot \frac{1}{6} = \frac{1}{2}$, which is also information theoretically optimal for this network.

Without using spatial reuse, for zero outage, the achievable normalized communication load (by the file size) such that every user can successfully decode is $R = \frac{M}{S}\left(1 - \frac{S}{M}\right) = \frac{M}{S} - 1$,[4] which is surprisingly almost the same as the result shown by [46], where instead of D2D communications, one central server (base station) which has access to all the video files, multicasts coded messages.

Suppose that (M, R) is achievable and that under the worst-case scenario, there exists a transmission policy that can deliver all coded messages necessary to decode the requested file of each user in no more than D channel uses. Then, per-user the throughput, measured in useful information bits per channel use, is given by $T = \frac{F}{D}$, where F denotes the file size in bits. Hence, we can see that the per-user throughput $\Theta\left(\frac{S}{M}\right)$ has the same scaling law as the throughput by using the decentralized random caching and uncoded delivery scheme described in Sec. 3. In addition, it can be shown that there is no further gain when spatial reuse is also exploited. In other words, *the gains of spatial reuse and coded delivery cannot accumulate*. Intuitively, if spatial reuse is not allowed, a complicated caching scheme can be designed such that one transmission can

[4] It can be seen that the per-user throughput T is inversely proportional to R under protocol model.

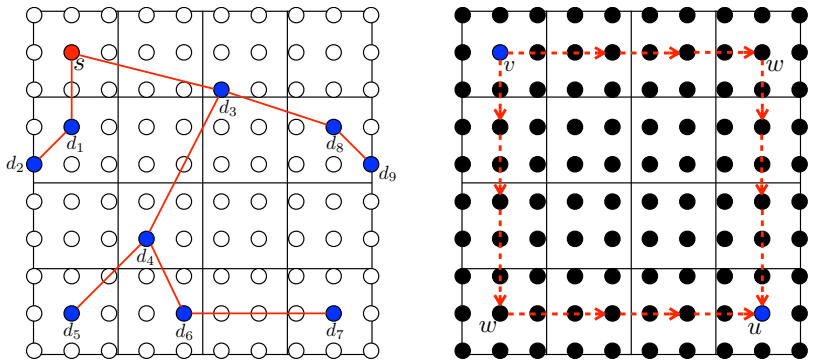
be useful for as many users as possible. If the transmission range is reduced and coding is done in one cluster, then the number of users benefited by one transmission is reduced but the D2D transmissions can operate simultaneously at a higher rate. Further, the complexity of caching subpacketization and coding can also be reduced. Therefore, the benefit of coding depends on the actual physical layer throughput (bits/s/Hz) and the caching/coding complexity rather than throughput scaling laws.[5]

The main drawback of the deterministic caching placement in the above proposed scheme is that, in practice, a tight control on the users caches must be implemented in order to make sure that, at any point in time, the files subpackets are stored into the caches in the required way. While this is conceptually possible, such approach is not robust to events such as user mobility and nodes turning on and off, as it may happen in a D2D wireless network with caching in the user devices. In [29], the authors proposed a decentralized random caching and coded delivery scheme that allows for more system robustness. In this scheme, each file of F bits is divided into L blocks or packets of F/L bits each. These packets are interpreted as the elements of the binary extension field $\mathbb{F}_{2^{F/L}}$, and are encoded using a $(L, L/\rho)$-MDS code, for some $\rho < 1$ the choice of which is essential to guarantee enough coded symbols per file are cached in the network such that the entire library is cached with high probability as $L \to \infty$. Notice that this expands the size of each packet from F to F/ρ. The resulting L/ρ encoded blocks of F/L bits each will be referred to as "MDS-coded symbols". Each user independently of the other users, uniformly chooses $\frac{ML}{m}$ elements from $\{1, 2, \cdots, L/\rho\}$ at random. In the delivery, similar to [45], each user seeks for possible coded multicasting opportunities with different group size. It can be shown that the achievable per user throughput of the proposed decentralized D2D coded caching scheme is the same as the deterministic case.

5.2 An Improved Coded D2D Caching Schemes

The scheme in [29] can be understood as dividing the D2D caching network into N independent shared-link caching networks, in each of which one device serves as the base station and the rest $N-1$ devices are users. In each of the independent shared-link caching network, the coded multicasting scheme proposed in [46] is applied. In [68], an improved coded D2D caching scheme is proposed. In this scheme, under the same cache placement as in [29,46], the delivery phase also divides the D2D network into N shared link caching networks. However, instead of applying the coded multicasting in each shared-link caching network in [46], it applies the *leader-based* coded multicasting scheme proposed in [69]. Consider a shared-link caching network with user demand vector $\mathbf{d}$ and denote $\mathcal{N}_e(\mathbf{d})$ as the number of distinct files in a request vector $\mathbf{d}$, in the *leader-based* coded multicasting scheme, any subset of users that request $\mathcal{N}_e(\mathbf{d})$ distinct files is referred to as leaders and the base station only needs to multicast the codewords involving the requests of the leaders. Based on this scheme, the improved

[5] An extensive analysis of the performance for D2D multicast is given by [43].



(a) (b)

Figure 6: (a) The D2D caching network is partitioned into square clusters with side length of $\frac{r}{\sqrt{5}}$. The red node s represents the source node and all the blue nodes represent its $t = 9$ multicast destinations $d_1, \cdots, d_9$. The constructed Euclidean Minimum Spanning Tree (EMST) for this multicast session is shown by the red, blue nodes and the red solid lines. (b) A illustration of the Manhattan routing approach from node v (transmitter) to node u (receiver). The red dashed line represent two manhattan routing paths. In this example, the transmission range is assume to be $\frac{3}{\sqrt{N}}$.

worst-case communication load for the coded D2D caching scheme is given by

$$R_{\text{Improved}} = \max_{\mathbf{d}} \frac{\binom{N-1}{t} - \frac{1}{N}\sum_{i=1}^{N}\binom{N-1-\mathcal{N}_e(\mathbf{d}\setminus\{i\})}{t}}{\binom{N-1}{t-1}}, \tag{16}$$

where $t = NS/M$ and $\mathbf{d} \setminus i$ denote all the elements in $\mathbf{d}$ except of i. It can be shown that when $N > M$, R_{Improved} strictly reduces the communication load in [29], which is $R = \frac{M}{S}\left(1 - \frac{S}{M}\right) = \frac{\binom{N-1}{t}}{\binom{N-1}{t-1}}$. Moreover, it can also be shown that R_{Improved} is optimal under the constraint of uncoded cache placement and one-shot delivery.[6]

5.3 Extension to Multihop Coded D2D Caching Networks

In Sec. 4, under uncoded cache placement and delivery constraint, we extended the uncoded D2D single-hop transmission scheme to the multihop case. For small outage probability, the per-user throughput of multihop delivery can be improved to $\Theta\left(\sqrt{\frac{S}{M}}\right)$, which shows an order gain compared to that for single-hop D2D communications. In Sec. 5.1, we allow coding for both cache placement phase, delivery (transmission) phase but focus on only single hop transmissions. Therefore, it is natural to consider the extension that relaxes all the constraints

[6]One-shot delivery means that each user can recover each of its needed bits from its own cache and the transmission of a single other device.

on cache placement, delivery and transmission phases under protocol channel model. It means that we can allow coding on both cache placement and delivery phases, and also allow multihop communications. Ref. [32] considered this scenario and proposed an intra-file coded cache placement based on deterministic assignment of Minimum Distance Separable (MDS)-coded packets of the files, a coded multicast delivery scheme where the users send linearly coded messages (similar to the example in Fig. 5) to each other in order to collectively satisfy their demands, and a randomized Euclidean Minimum Spanning Tree (EMST) based routing strategy for transmission, where each hop in the EMST based routing is implemented via the Manhattan Routing approach. These are illustrated in Fig. 6. Under the worst-case demands, with some constant C, it shows that this approach actually achieves the throughput scaling law of $T = C\frac{\sqrt{\frac{S}{M}}}{1-\frac{S}{M}} = \Theta\left(\sqrt{\frac{S}{M}}\right)$, which has the same scaling law as the per-user throughput achieved by the multihop D2D uncoded scheme under the PPP model. This throughput scaling law in [32] also achieves information theoretic outer bound within a multiplicative constant factor in practical parameter regimes.

Moreover, the achievable throughput in the multi-hop D2D coded caching can be written as the product of four terms as follows.

$$T = C\frac{\sqrt{\frac{S}{M}}}{1-\frac{S}{M}} = \Theta\left(\frac{1}{N}\cdot\frac{1}{1-\frac{S}{M}}\cdot t\cdot\sqrt{\frac{N}{t}}\right), \tag{17}$$

where $t = \frac{NS}{M}$ is the ratio between the aggregate memory in the network and the library size. We have the following interpretation for (17). The first three terms $\frac{1}{N}$, $\frac{1}{1-\frac{S}{M}}$ and t are also found in the shared link caching networks [46] and the single-hop D2D caching networks [29]. In particular, the term $\frac{1}{N}$ is the per-user throughput by using a conventional scheme that serves individual demands without exploiting the demand redundancy; $\frac{1}{1-\frac{S}{M}}$ can be viewed as the *local caching gain*, any user can cache a fraction S/M of each file, hence it needs to receive only the remaining part; t is referred to as *global caching gain*, which is the gain due to the aggregate memory in the network rather than individual user's memory.[7] The new term $\sqrt{\frac{N}{t}}$ is the *multihop transmission gain*, which is the gain of multhop D2D caching networks over the single-hop D2D caching networks and obtained by using multihop. Intuitively, this term can be interpreted as follows. There are at most $\Theta(N)$ concurrent transmissions in the network. Each user has t destinations uniformly selected at random such that the average number of hops per bit is $\Theta(\sqrt{Nt})$.[8] Hence, the average number of bits that can be sent in the network per time slot is given by $\Theta\left(\frac{N}{\sqrt{Nt}}\right) =$

[7]Note that the global caching gain for the shared link caching network is $t+1$ [46]. For $NS \gg M$ ($t \geq 1$), these factors are almost identical.

[8]Note that for unicast network, where $t = 1$, the number of hops per bit reduces to $\Theta(\sqrt{n})$ as shown in the seminal paper [22].

$\Theta\left(\sqrt{\frac{N}{t}}\right)$.

Although the proposed D2D coded multi-hop caching scheme does not gain in terms of the per-user throughput, it may have some other gains. For example, the amount of data traffic N_S generated at each node from its memory is given by

$$\begin{aligned} N_S &= \frac{F}{(1-\varepsilon)t\binom{N}{t}} \cdot \binom{N-1}{t} \\ &= \frac{F}{1-\varepsilon}\left(1-\frac{S}{M}\right)\frac{1}{t} = \Theta\left(\left(1-\frac{S}{M}\right)\frac{F}{t}\right), \end{aligned} \tag{18}$$

where ε is an arbitrarily small constant. In contrast, due to the constraint of caching the entire files, the data traffic generated at each user from its own memory for the scheme discussed in Sec. 4 is given by $\Theta(F)$ bits for fixed S/M. Hence, in terms of the data generated at each node, the proposed scheme in this paper has a gain of t, which is the global caching gain.

6 Practical Implementation Considerations for D2D Caching Networks

In this section, we will discuss some practical consideration regarding the implementations of D2D caching networks. In particular, we will first discuss D2D caching under realistic channel conditions. Second, we will focus on the mobile D2D caching networks. Finally, we will discuss the complexity issues regarding D2D coded caching.

6.1 D2D caching under realistic channel model

The results based on protocol channel model developed in previous sections reveal the potential of D2D caching networks in a fundamental manner without much practical considerations. In the literature, there are extensive studies on relaxing the constraints of the simplified protocol channel model via the consideration of much more realistic physical layer channel models. It turns out that coupling the realistic physical layer channel models significantly increase the difficulty in terms of theoretical analysis. Hence, one needs to find good models that can capture the main features of wireless D2D caching networks and are analytically tractable. A significant amount of work in this domain has been developed in the past few years, see [1,3,7,15,35,47,52,55,61,67] for a few examples.

When interferences are treated as noise, in general, mainly three representative models are used. The first two models consider the sub-6G Hz channel model and the location of the devices are based on various Poisson Point Processes (PPPs) and Poisson Cluster Processes (PCPs), where the former does not consider the correlation of the user devices while the latter considers this

phenomenon. The third model focuses on the mm-wave channel model, which is very attractive for D2D transmissions due to the high available bandwidth. The location distributions of the devices also follow PPPs. For each model, due to the difficulty of analysis, rather than the scaling laws of throughput-outage tradeoff , different performance metrics including Density of Successful Receptions (DSR) [47], Coverage Probability, Area Spectral Efficiency (ASE) [2] and Offloading Factor [15] are defined and analyzed. Although the caching polices can be different among distinct physical layer models, the analysis and simulation results using these models confirm the significant advantages of D2D caching networks in terms of different throughput related metrics.

When interferences are not treated as noise, or cooperative D2D communication is used, it turns out that cached information can be used to mitigate interference. In [21], it assumed a random phase fading proposed a delivery scheme based on the Hierarchical MIMO Cooperation scheme originally proposed in [56]. Let all nodes be uniformly and independently distributed in a unit square, let $SN > M$, the network is divided into square clusters of area $n^{-\nu}$ for some $\nu > 0$. The user requests follows a uniform distribution (Zipf distribution with $\gamma = 0$). The cache placement is similar to the random decentralized cache placement that discussed in Sec. 5.1, where no MDS coding is used ($\rho = 1$). In the delivery phase, each user will first identify L users holding distinct packets requested by this user as the source nodes. Then, to serve the targeted user, the hierarchical MIMO cooperation scheme in [56] can be applied. In particular, a virtual MIMO transmission can be formed from these L source nodes to the nodes in the corresponding cluster of the targeted user. Then each user in the corresponding cluster quantizes the received signal and sends it to the targeted user which will jointly process the all copies of superimposed signals received from previous virtual MIMO transmissions. For serving all users in the network, the same transmission policy is applied in a TDMA manner. The per-user throughput of this scheme is $\Theta\left(n^{-\frac{1}{\tau+1}}\right)$ with a vanishing outage probability, where τ is an integer depending on the number of hierarchies but independent of N. It can be seen that in this case, the per-user throughput improves significantly compared to that for both single-hop and multihop D2D communications under protocol model (interference avoidance). As an indpendent work, Ref. [44] considered the extended network model with the similar physical layer assumptions.[9] In particular, a hierarchical cache content placement based on the content popularity and a tree-graph-based content delivery based on both multihop and hierarchical MIMO cooperation are proposed. Under the same extended network condition and a Zipf popularity distribution, it can be shown that this proposed scheme achieves zero outage probability and has an order gain in terms of per user throughput when the path loss parameter is less than 3 and the Zipf parameter $\gamma < 3/2$.

From the previous discussions, it can be seen that there is indeed no order

[9] All model considered in this chapter is a dense network model, whose area is a constant while the area of extended networks can grow linearly with the number of users and the nearest distance between two users are above a constant.

gain in terms of throughput scaling laws of D2D coded caching in both single-hop and multi-hop compared to the counterpart of D2D uncoded caching schemes. The achievable throughput of D2D caching in practice will depends on the realistic physical layer channel conditions.

6.2 D2D Caching in Mobile Networks

The previous sections assumed all D2D users are static, or do not assume explicit models for their mobility. In this section, we discuss how mobility can affect the behavior and the performance of D2D caching networks.

The effect of user mobility on D2D uncoded caching is investigated via simulations in [18], which shows that user mobility does not have a significant impact on a random caching scheme. In addition, as discussed in Sec. 5.1, in [29], a decentralized coded caching scheme based on random cache placement is proposed and this scheme can also be robust for user mobility since the distribution of cache placement may not change when user moves. However, such caching schemes may not take full advantages of the specific user mobility pattern. The authors in [37] show that user mobility has positive effect on D2D caching and the authors in [38] consider the case where mobile users can update cache placement based on the demand and user mobility. However, it is assumed that one complete file can be transmitted via any D2D link when two users contact, which may not be not practical. A random work based user mobility model is introduced in [25], where user moves follow a discrete Markov process. Furthermore, another interesting user mobility models based on the distribution of contact time/inter-contact time is introduced in [25,62], which will be discussed in the following.

In this model, mobile users may have contact with each other when they are within the transmission range. Correspondingly, the *contact time* for two mobile users is defined as the time slot that they are able to serve each other. The duration of each contact of user i and j is $t^c_{i,j}$ seconds. The *inter-contact time* for two mobile users is defined as the time duration between two consecutive contact times. In particular, the locations of contact times in the timeline for any two users i and j are modeled as a Poisson process with intensity $\lambda_{i,j}$, which represent the average number of contacts per unit time slot. For simplicity, the timelines for different user pairs are independent. Similar to the decentralized D2D coded caching discussed in Sec. 5.1, an MDS type coded file cache placement is considered. A stochastic mixed integer nonlinear programming with the consideration of the contact-time and inter-contact time of mobile users is formulated with average data offloading ratio as the objective function and the cache size as the constraint. Sub-optimal solutions based on dynamic programming and sub-modular optimization are evaluated. Simulation results show the significant advantage of taking the consideration of the mobility pattern compared to the commonly used caching strategies including popular caching and random caching placement. Interestingly, it can be observed that users with very low velocity tend to rarely be in contact with others, and thus they need to cache the most popular files to meet their own requests. On the other hand,

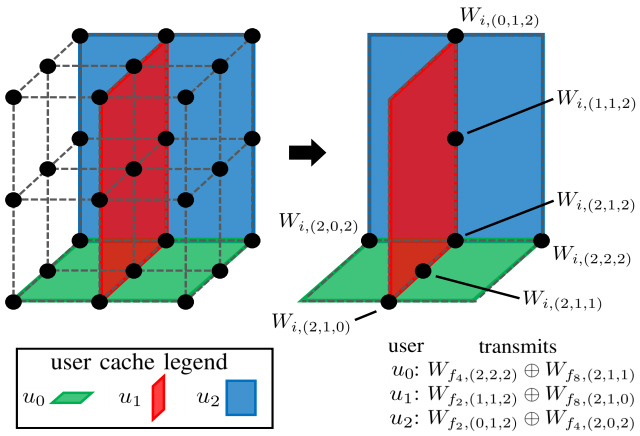


Figure 7: Each point on the 3D-lattice represents a set of packets. Each user's cache is represented by a plane of lattice points. The right figure represents a multicasting group, or a collection of 3 planes, where any two are orthogonal. The points on the intersection of two or more of these planes are highlighted. Every two users in this group share two packets that are requested by the third user. $W_{i,(j,k,\ell)}, i \in \{1,\ldots,M\}, j,k,\ell \in \{0,1,2\}$ represent the packets of file i. In this example, $N = 9$ and $t = 3$. The required number of packets per file is 27. The required number of packets using the scheme in [29] is 252.

the users with high velocity contact others frequently, and they need to cache the most popular files as well, in order to help others to download the requested files. This work is generalized by the same authors in [63], which relaxes the assumption of constant contact durations and provides analytical performance evaluations.

6.3 Complexity of D2D Coded Caching

When D2D coded caching is considered, one of the most significant practical constraints is the requirement of extremely large subpacketization levels. This will lead to high complexity of D2D coded caching schemes. For example, in [29], each file needs to be partitioned into $\binom{N}{t}t$ subfiles, where $t = \frac{NS}{M}$ represents the aggregate storage capacity in the network. When $\mu = \frac{S}{M}$ is fixed, $\binom{N}{t}t = t2^{NH_b(\mu)}$ grows exponentially with n. To address this concern, Refs. [64–66] proposed several approaches to design coded caching networks with reduced subpacketization levels. In particular, two combinatorial design approaches were proposed for centralized D2D caching networks which have reduced subpacketization compared to [29]. The first approach is based on a "hypercube" design to define the cache placement. It demonstrates how the geometry of this hypercube can help to exploit coded multicasting opportunities for the delivery. One example of the 3-dimension hypercube (or cube) with 9 users and $t = 3$ is shown in Fig. 7. It can be observed that the hypercube

approach is designed and optimized specifically for D2D caching networks as opposed to adapting an existed caching scheme designed for shared-link caching network as the approach proposed in [32]. The number of required subfiles per file is reduced to $\left(\frac{M}{S}\right)^t$ from $\binom{N}{t}t$ while the communication load is $\frac{M}{S}$ which is almost the same as $\frac{N}{S}-1$ as in [29]. In addition, by adopting the idea recently proposed in [33], this scheme can also be extended to a decentralized coded D2D caching scheme, which leads to a much more flexible design for given network parameters. Meanwhile, the advantage of the reduced subpacketizations of the hypercube approach still remains in the decentralized designs of the D2D caching networks. The second approach is based on an application of the Ruzsa-Szeméredi graph proposed in [6, 58], which is firstly used for the design in shared-link caching networks in [60]. Ref. [64] applied the Ruzsa-Szeméredi graph based design in coded D2D caching networks and show that the requirement of file subpacketization is at most sub-quadratic in terms of the number of users if no spatial reuse is allowed while the per-user throughput scales as $\Theta(N^{-\delta})$ for some arbitrarily small δ under some parameter regimes. Both D2D combinatorial designs sustain the significant throughput gain compared to conventional uncoded unicast [30] and the required packetizations are reduced exponentially compared to [29] with respect to the number of users N while keeping the library size M and memory size S fixed. Finally, the impact of enabling spatial reuse in these caching network designs is also studied and show this can further reduce the packetizations levels, while also improving the per-user throughput significantly for some parameter regimes, in contrast to the case in [29]. All these work show gives a hint that although the D2D caching schemes can be transferred from the shared-link caching schemes without much loss in terms of throughput as shown in [29], when subpacketizations are concerned, the D2D caching networks need a specific design approach that is different from the shared-link caching network. This idea is formalized in [71], in which the authors shows that under some parameter regimes for D2D caching networks, there indeed exists a different design that achieves the exact optimal communication load as in [29] while requiring an order-wise smaller number of packets per file.

7 Conclusions and Further Readings

The area of D2D caching networks has attracted significant attentions in recent years. In this chapter, we discussed a few fundamental approaches that are typical for analyzing the scaling laws of D2D caching networks including single-hop D2D uncoded/coded caching and multi-hop D2D unocded/coded caching approaches. Tight scaling laws are characterized and show great potential of D2D coded caching network. In particular, the per-user throughput gain of caching is multiplicative under different network models.

There are a number of other important and interesting directions for this topic. We will illustrate a few of them as follows. First, under the condition of treating interference as noise, unlike the work discussed in Sec. 6.1, where

only simple scheduling schemes are considered, Ref. [52] applied the ITLinQ scheduling approach [51], which introduced the optimal condition of treating interference as noise. Second, it is also possible to align or cancel interference, or use cooperation approaches rather than treating interference as noise. For example, Refs. [23, 53] focused on the interference channel,[10] and used zero forcing and cache cancellation or interference alignment to remove the effect of interference. Finally, in the area of coded D2D caching, besides the paper discussed in Sec. 5.1, there are other coded D2D caching approaches such as [26] for distinct cache sizes and [70] for secure D2D caching.

References

[1] M. Afshang and H. S. Dhillon. Optimal geographic caching in finite wireless networks. *arXiv preprint arXiv:1603.01921*, 2016.

[2] M. Afshang, H. S. Dhillon, and P. H. J. Chong. Fundamentals of cluster-centric content placement in cache-enabled device-to-device networks. *IEEE Transactions on Communications*, 64(6):2511–2526, June 2016.

[3] M. Afshang, H. S. Dhillon, and P. H. Joo Chong. Modeling and performance analysis of clustered device-to-device networks. *IEEE Transactions on Wireless Communications*, 15(7):4957–4972, July 2016.

[4] A. Agarwal and P. R. Kumar. Capacity bounds for ad hoc and hybrid wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(3):71–81, 2004.

[5] I. Ahmed, M. H. Ismail, and M. S. Hassan. Video transmission using device-to-device communications: A survey. *IEEE Access*, 7:131019–131038, 2019.

[6] N. Alon, A. Moitra, and B. Sudakov. Nearly complete graphs decomposable into large induced matchings and their applications. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1079–1090. ACM, 2012.

[7] R. Amer, H. Elsawy, M.M. Butt, E.A. Jorswieck, M. Bennis, and N. Marchetti. Optimizing joint probabilistic caching and communication for clustered d2d networks. *arXiv preprint arXiv:1810.05510*, 2018.

[8] Bartlomiej Blaszczyszyn and Anastasios Giovanidis. Optimal geographic caching in cellular networks. In *2015 IEEE international conference on communications (ICC)*, pages 3358–3363. IEEE, 2015.

[10] Note that the interference channel can be understood as D2D communications when the transmitters and receivers are fixed.

[9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999.

[10] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. *ACM SIGCOMM Computer Communication Review*, 32(4):177–190, 2002.

[11] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal throughput–delay scaling in wireless networks—part ii: Constant-size packets. *IEEE Transactions on Information Theory*, 52(11):5111–5116, 2006.

[12] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Optimal throughput-delay scaling in wireless networks-part i: The fluid model. *IEEE Transactions on Information Theory*, 52(6):2568–2592, 2006.

[13] M. Franceschetti, O. Dousse, D.N.C. Tse, and P. Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *Information Theory, IEEE Transactions on*, 53(3):1009–1018, 2007.

[14] A El Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *IEEE INFOCOM 2004*, volume 1. IEEE, 2004.

[15] N. Giatsoglou, K. Ntontin, E. Kartsakli, A. Antonopoulos, and C. Verikoukis. D2d-aware device caching in mmwave-cellular networks. *IEEE Journal on Selected Areas in Communications*, 35(9):2025–2037, Sep. 2017.

[16] S. Gitzenis, GS Paschos, and L. Tassiulas. Asymptotic laws for joint content replication and delivery in wireless networks. *Arxiv preprint arXiv:1201.3095*, 2012.

[17] N. Golrezaei, A. G. Dimakis, and A. F. Molisch. Scaling behavior for device-to-device communications with distributed caching. *IEEE Transactions on Information Theory*, 60(7):4286–4298, July 2014.

[18] N. Golrezaei, P. Mansourifard, A.F. Molisch, and A.G. Dimakis. Base-station assisted device-to-device communications for high-throughput wireless video networks. *Wireless Communications, IEEE Transactions on*, 13(7):3665–3676, July 2014.

[19] N. Golrezaei, A.F. Molisch, A.G. Dimakis, and G. Caire. Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution. *Communications Magazine, IEEE*, 51(4):142–149, 2013.

[20] Negin Golrezaei, Alexandros G Dimakis, Andreas F Molisch, and Giuseppe Caire. Wireless video content delivery through distributed caching and peer-to-peer gossiping. In *Signals, Systems and Computers (ASILOMAR),*

2011 Conference Record of the Forty Fifth Asilomar Conference on, pages 1177–1180. IEEE, 2011.

[21] J. Guo, J. Yuan, and J. Zhang. An achievable throughput scaling law of wireless device-to-device caching networks with distributed mimo and hierarchical cooperations. *IEEE Transactions on Wireless Communications*, 17(1):492–505, Jan 2018.

[22] P. Gupta and P.R. Kumar. The capacity of wireless networks. *Information Theory, IEEE Transactions on*, 46(2):388–404, 2000.

[23] J. Hachem, U. Niesen, and S. N. Diggavi. Degrees of freedom of cache-aided wireless interference networks. *IEEE Transactions on Information Theory*, 64(7):5359–5380, 2018.

[24] M. Hefeeda and O. Saleh. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Transactions on Networking*, 16(6):1447–1460, December 2008.

[25] S. Hosny, A. Eryilmaz, A. A. Abouzeid, and H. El Gamal. Mobility-aware centralized d2d caching networks. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 725–732, Sept 2016.

[26] A. M. Ibrahim, A. A Zewail, and A. Yener. Device-to-device coded caching with distinct cache sizes. *arXiv preprint arXiv:1903.08142*, 2019.

[27] S. W. Jeon, S. N. Hong, M. Ji, G. Caire, and A. F. Molisch. Wireless multihop device-to-device caching networks. *IEEE Transactions on Information Theory*, 63(3):1662–1676, March 2017.

[28] M. Ji, G. Caire, and A. F. Molisch. The throughput-outage tradeoff of wireless one-hop caching networks. *IEEE Transactions on Information Theory*, 61(12):6833–6859, Dec 2015.

[29] M. Ji, G. Caire, and A. F. Molisch. Fundamental limits of caching in wireless d2d networks. *IEEE Transactions on Information Theory*, 62(2):849–869, Feb 2016.

[30] M. Ji, G. Caire, and A. F. Molisch. Wireless device-to-device caching networks: Basic principles and system performance. *IEEE Journal on Selected Areas in Communications*, 34(1):176–189, Jan 2016.

[31] M. Ji, G. Caire, and A.F. Molisch. Fundamental limits of distributed caching in d2d wireless networks. In *Information Theory Workshop (ITW), 2013 IEEE*, pages 1–5. IEEE, 2013.

[32] M. Ji, R. Chen, G. Caire, and A. F. Molisch. Fundamental limits of distributed caching in multihop d2d wireless networks. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2950–2954, June 2017.

[33] Sian Jin, Ying Cui, Hui Liu, and Giuseppe Caire. New order-optimal decentralized coded caching schemes with good performance in the finite file size regime. *arXiv preprint arXiv:1604.07648*, 2016.

[34] L.-S. Juhn and L.-M. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcast.*, 43(3):268–271, September 1997.

[35] R. Karasik, O. Simeone, and S. Shamai. Information-theoretic analysis of d2d-aided pipelined content delivery in fog-ran. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–5. IEEE, 2018.

[36] J. Karedal, A. J. Johansson, F. Tufvesson, and A. F. Molisch. A measurement-based fading model for wireless personal area networks. *IEEE Transactions on Wireless Communications*, 7(11):4575–4585, November 2008.

[37] S. Krishnan and H. S. Dhillon. Effect of user mobility on the performance of device-to-device networks with distributed caching. *IEEE Wireless Communications Letters*, 6(2):194–197, April 2017.

[38] R. Lan, W. Wang, A. Huang, and H. Shan. Device-to-device offloading with proactive caching in mobile cellular networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015.

[39] M.-C. Lee, M. Ji, and A. F. Molisch. Optimal throughput–outage analysis of cache-aided wireless multi-hop D2D networks – Derivations of scaling laws. *arXiv preprint arXiv:2005.05149*, May 2020.

[40] M.-C. Lee, M. Ji, A. F. Molisch, and N. Sastry. Throughput-outage analysis and evaluation of cache-aided d2d networks with measured popularity distributions. *IEEE Transactions on Wireless Communications*, 18(11):5316–5332, November 2019.

[41] L. Li, G. Zhao, and R. S. Blum. A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies. *IEEE Communications Surveys & Tutorials*, 20(3):1710–1732, 2018.

[42] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk. An overview of 3gpp device-to-device proximity services. *IEEE Communication Magazine*, 52(4):40–48, 2014.

[43] X. Lin, R. Ratasuk, A. Ghosh, and J. G. Andrews. Modeling, analysis and optimization of multicast device-to-device transmissions. 2013.

[44] A. Liu, V. K. N. Lau, and G. Caire. Cache-induced hierarchical cooperation in wireless device-to-device caching networks. *IEEE Transactions on Information Theory*, 64(6):4629–4652, June 2018.

[45] M. A. Maddah-Ali and U. Niesen. Decentralized caching attains order-optimal memory-rate tradeoff. *arXiv preprint arXiv:1301.5848*, 2013.

[46] M. A. Maddah-Ali and U. Niesen. Fundamental limits of caching. *Information Theory, IEEE Transactions on*, 60(5):2856–2867, 2014.

[47] D. Malak, M. Al-Shalash, and J. G. Andrews. Optimizing content caching to maximize the density of successful receptions in device-to-device networking. *IEEE Transactions on Communications*, 64(10):4365–4380, Oct 2016.

[48] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek. Device-enhanced mec: Multi-access edge computing (MEC) aided by end device computation and caching: A survey. *IEEE Access*, 7:166079–166108, 2019.

[49] A. F. Molisch, G. Caire, D. Ott, J. R. Foerster, D. Bethanabhotla, and M. Ji. Caching eliminates the wireless bottleneck in video aware wireless networks. *Adv. Elect. Eng.*, 2014(261390), November 2014.

[50] A.F. Molisch. *Wireless communications*. 2nd edition, IEEE Press - Wiley, 2011.

[51] N. Naderializadeh and A. S. Avestimehr. Itlinq: A new approach for spectrum sharing in device-to-device communication systems. *IEEE Journal on Selected Areas in Communications*, 32(6):1139–1151, June 2014.

[52] N. Naderializadeh, D. T. H. Kao, and A. S. Avestimehr. How to utilize caching to improve spectral efficiency in device-to-device wireless networks. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 415–422, Sept 2014.

[53] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr. Fundamental limits of cache-aided interference management. *IEEE Transactions on Information Theory*, 63(5):3092–3107, 2017.

[54] H. Nishiyama, M. Ito, and N. Kato. Relay-by-smartphone: realizing multihop device-to-device communications. *IEEE Communication Magazine*, 52(4):56–65, 2014.

[55] T. Nuradha, T. Samarasinghe, and K.T. Hemachandra. A novel content caching and delivery scheme for millimeter wave device-to-device communications. *arXiv preprint arXiv:1911.06517*, 2019.

[56] A. Ozgur, O. Leveque, and D. N. C. Tse. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Transactions on Information Theory*, 53(10):3549–3572, Oct 2007.

[57] L. Qiu and G. Cao. Popularity-aware caching increases the capacity of wireless networks. *IEEE Transactions on Mobile Computing*, 19(1):173–187, 2019.

[58] I. Ruzsa and E. Szemerédi. Triple systems with no six points carrying three triangles. *Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai*, 18:939–945, 1978.

[59] Y. Sánchez de la Fuente, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Le Louédec. idash: improved dynamic adaptive streaming over http using scalable video coding. In *Proceedings of the second annual ACM conference on Multimedia systems*, pages 257–264. ACM, 2011.

[60] K. Shanmugam, A. M. Tulino, and A. G. Dimakis. Coded caching with linear subpacketization is possible using ruzsa-szem\'eredi graphs. *arXiv preprint arXiv:1701.07115*, 2017.

[61] S. Vuppala, T. X. Vu, S. Gautam, S. Chatzinotas, and B. Ottersten. Cache-aided millimeter wave ad hoc networks with contention-based content delivery. *IEEE Transactions on Communications*, 66(8):3540–3554, Aug 2018.

[62] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief. Mobility-aware caching in d2d networks. *IEEE Transactions on Wireless Communications*, 16(8):5001–5015, Aug 2017.

[63] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief. Exploiting mobility in cache-assisted d2d networks: Performance analysis and optimization. *IEEE Transactions on Wireless Communications*, 17(8):5592–5605, Aug 2018.

[64] N. Woolsey, R. Chen, and M. Ji. Device-to-device caching networks with subquadratic subpacketizations. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.

[65] N. Woolsey, R. Chen, and M. Ji. Coded caching in wireless device-to-device networks using a hypercube approach. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, May 2018.

[66] N. Woolsey, R.-R. Chen, and M. Ji. Towards practical file packetizations in wireless device-to-device caching networks. *arXiv preprint arXiv:1712.07221*, 2017.

[67] W. Wu, N. Zhang, N. Cheng, Y. Tang, K. Aldubaikhy, and X. Shen. Beef up mmwave dense cellular networks with d2d-assisted cooperative edge caching. *IEEE Transactions on Vehicular Technology*, 68(4):3890–3904, April 2019.

[68] Ç. Yapar, K. Wan, R. F. Schaefer, and G. Caire. On the optimality of d2d coded caching with uncoded cache placement and one-shot delivery. *IEEE Transactions on Communications*, 67(12):8179–8192, Dec 2019.

[69] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr. The exact rate-memory tradeoff for caching with uncoded prefetching. *IEEE Transactions on Information Theory*, 64(2):1281–1296, Feb 2018.

[70] A. A. Zewail and A. Yener. Device-to-device secure coded caching. *IEEE Transactions on Information Forensics and Security*, 15:1513–1524, 2020.

[71] X. Zhang, X.F Yang, and M. Ji. A new design framework on device-to-device coded caching with optimal rate and significantly less subpacketizations. *to appear in IEEE ISIT 2020*, 2020.