Caching in Large Wireless Networks

Georgios Paschos and Marios Kountouris

I. Introduction

With more than one billion wireless subscribers today and predictions for this number being tripled over the next five years, the ICT industry is confronted with an avalanche of traffic demand. The unprecedented traffic increase of wireless network traffic is a side effect of the info-revolution we are experiencing, fueled by the proliferation of smart mobile devices and bandwidth-greedy video applications. The quarterly Internet monitoring report of Akamai reveals that the network traffic has reached the exa-byte scale and keeps increasing steadily. Reports from Cisco indicate that mobile traffic gets a larger and larger share of the pie, soon reaching 60% [1].

A well-established fundamental result states that long multihop wireless communication do not scale [2], i.e., the maximum common traffic rate for all flows is inversely proportional to the average number of hops. As a result, the proposed techniques to tackle the communication challenges of future networks focus mostly on further enhancing the link capacities and densifying the wireless network with more access points (APs) or base stations (BSs), which are connected to Internet with wires. Several advanced transmission technologies have been proposed and implemented in order to boost the performance of wireless networks. These advanced technologies include the aggregation and concurrent use of different frequency carriers, the exploitation of the spatial dimension using multi-antenna (MIMO) techniques, and the cooperative communication where multiple nodes are able to coordinate their scheduling or transmission to serve users with adverse channel conditions. Despite continuous progress, all these cutting edge physical layer (PHY) technologies do not allow significant enhancements as they are reaching their theoretical limits. Ultra dense networks (UDNs) and heterogeneous networks (HetNets) are envisioned to provide a significant network performance leap by deploying heterogeneous infrastructure, e.g., remote radio heads (RRHs) and low power nodes such as picocells, small cells, femto APs, and relays. Bringing the network closer to end-users, the link quality can be enhanced due to the link reduced distance and the larger number of cells, which allows for more efficient spectrum reuse and therefore larger data rates. Nevertheless, in large-scale dense networks, backhaul connectivity, capacity and reliability may become the performance bottleneck, without even mentioning the additional cost associated with their deployment and maintenance.

Two decades ago, the World Wide Web was experiencing a similar challenge. Under the legacy client-server model, every browser in the world needed to connect to the same remote server. The sum of traffic of these end-to-end connections was creating a catastrophic increase of traffic in the Internet backhaul. The solution to that challenge was *web caching*; a technique borrowed from computing processors and operating systems, which exploits the temporal together with the spatial vicinity of user requests to improve the user-perceived Quality of Service (QoS) and network performance metrics. The exploitation of web caching led to the proliferation of Content Delivery Networks (CDNs), responsible for replicating content across the entire Internet today, and intercepting more than 60% of traffic that is directed to remote servers, thereby eliminating a significant amount of backhaul traffic.

With motivation from the success story of CDNs, it has been recently proposed to use caching to improve the sustainability of wireless networks [3], [4]. *Moving from CDNs to wireless networks does not simply produce a "wireless CDN"*. New paradigms of content replication appear, new schemes promise to improve efficiency by combining caching with properties of PHY layer, while new challenges are imposed by mobility, geometrical properties of wireless signal propagation, and the interference. Many longstanding models and the associated conventional wisdom may have to be revisited in the coming years.

In this chapter we study the fundamental theoretical underpinnings of wireless networks with caching capabilities. More specifically:

The authors are with Mathematical and Algorithmic Sciences Lab, France Research Center, Huawei Technologies Co., Ltd.
The opinions in this chapter are those of the authors only and not of the Huawei Technologies company.

1) We present an analytical tool for simplifying the calculations required to decide the optimal content replication in a wireless network. This tool is used to derive the asymptotic laws of backhaul capacity scaling for multihop wireless networks. The presented results consider (i) the number of networks nodes N, (ii) the number of contents M, and (iii) the cache size K as the key system "size" parameters that increase arbitrarily at different proportions. By studying the different cases we extract valuable intuition into the benefits of caching for the sustainability of wireless networks.
2) We investigate the performance of dense cache-enabled small cell networks and provide crisp insights on how the system operating values, the network geometry and interference affect the network performance. Using tools from stochastic geometry to model the node distribution, we provide design guidelines where to place the most popular content, i.e. at the mobile device or at the AP/BS. The effect of spatial correlation in content requests is analyzed.
3) The chapter is concluded with a summary of interesting research directions for the future that can play a crucial role in the proliferation of caching as a wireless network technique.

II. Sustainable Multihop Wireless Networks with Caching

The sustainability of wireless multihop networks can be characterized by means of studying a network that grows in size and examining its capacity scaling laws. In their seminal work [2], Gupta and Kumar studied the asymptotic behaviour of multihop wireless networks when communications take place between N pairs of randomly chosen nodes. In such a scenario the maximum data rate is $O(1/\sqrt{N})$, hence as the network grows ($N \to \infty$) the data rate vanishes to zero. This finding argues against the sustainability of multihop communications. Moreover, the $O(1/\sqrt{N})$ law was shown to arise from geometric considerations of the 2D wireless transmissions [5], hence it is impossible to breach under the classical random communicating pair model.

In this section we revisit the subject of the sustainability of wireless multihop networks with the added element of caching. Each node in the network represents a user that is interested in a particular content (instead of communicating with another node). The content can be replicated in the network caches, and thus the user can conviniently retrieve it from a nearby cache, reducing in this way the number of traversed hops. The derived asymptotic laws for capacity scaling suggest that caching has a powerful effect in the sustainability of wireless networks, and there exist interesting regimes where the $O(1/\sqrt{N})$ law is breached.

The fundamental size parameters are the number of nodes N, and the number of contents M, and thus the scaling laws will depend on how these increase to infinity. Another key parameter is the cache size K which depicts the number of contents that can be cached at each node. Taking K to infinity represents an interesting regime that reflects networks where nodes are upgraded as storage gets abundant and inexpensive [6]. Another influencing factor is the content popularity. With skewed popularity, popular files are requested multiple times and hence smaller K have a bigger impact on the system performance. As anticipated, in our analysis the popularity skewness crucially affects the sustainability of wireless networks.

The presented analysis is a summary of [7]–[10] and involves the solution of a hard combinatorial optimization that captures the best way of replicating contents with different popularities and retrieving them over network routes. Fortunately, by relaxing the problem to a convex optimization and rounding the solution, an order optimal performance can be obtained.

A. System Model

1) Network Model: The network is modelled by a grid topology as in Figure 1, which captures essential characteristics of the wireless networks (i) multihop short-range communications are preferable [2], and (ii) the network diameter scales as $\sqrt{N}$.[1] More realistic random topologies are handled in [12], but they are shown to result in the same scaling laws.

Formally, we consider a set of N nodes indexed by $n \in \mathcal{N} \triangleq \{1, 2, \ldots, N\}$, arranged on a $\sqrt{N} \times \sqrt{N}$ square grid on the plane. Each node is connected via undirected links to its four neighbours that lie next to it on the same row or column. By keeping the node density fixed and increasing the network size N, we obtain a scaling network

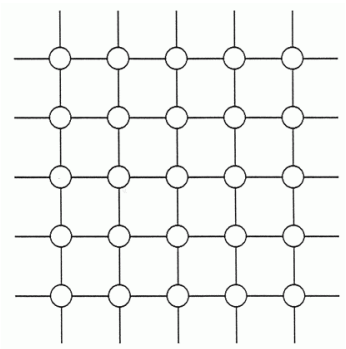

Fig. 1. For the scaling laws analysis, the network topology is considered to be a toroidal square grid.

similar to [2], with random pair throughput scaling as $O(1/\sqrt{N})$. Moreover, to avoid boundary effects, we consider a toroidal structure as in [5].

2) Content Requests: Nodes generate requests to a catalog of contents, indexed by $m \in \mathcal{M} \triangleq \{1, 2, \ldots, M\}$. Normally each node requests content m with rate λ_n^m. Such a consideration would lead the analysis of sustainability to the formation of the capacity region (which is the set of all rate vectors $[\lambda_n^m]$ that are sustainable), see [13], which however is expressed through a set of inequalities and will not provide a simple expression for the scaling laws. Hence to simplify the exposition and get a quantitive result, we assume the symmetric case where all nodes have $\lambda_n^m = \lambda^m, \ \forall n$.

More specifically, we consider the Independent Reference Model (IRM) [14], according to which the requests for file m are due to an independent Poisson process with intensity λp_m, where p_m represents the file popularity distribution–the latter is assumed time-invariant.

Last, we fix $\lambda = 1$ and study the required link capacity to support the communications. This is the inverse of the classical approach where we fix the link capacity and study the scaling of maximum supportable throughput. For example, in a network where under fixed link capacities the throughput scales as $O(1/\sqrt{N})$, our model will yield required link capacity scaling of $\Theta(\sqrt{N})$ to support constant throughput $\lambda = 1$. More generally, the obtained scaling laws for required link capacity can be inverted to reflect the throughput scaling laws. Accordingly, for sustainability we require the link capacity to be as small as possible, ideally to be $O(1)$.

3) Caching Model: Each node n is equipped with a cache, whose contents are denoted by the set $\mathcal{B}_n$, a subset of $\mathcal{M}$. If a request at node n regards a file m that lies in $\mathcal{B}_n$, then it is served locally. Due to the limited memory space of the cache, m will often be not available in $\mathcal{B}_n$, thus, node n will have to request m over the network from some other node w that keeps m in its cache. We denote with $[\mathcal{R}_{n,k}]$ the set of routes connecting nodes to nearby caches. The choice of the sets $[\mathcal{B}_n]$ and $[\mathcal{R}_{n,k}]$ crucially affects the network link loading.

Let, moreover, K be the storage capacity of nodes' cache measured in the number of files it can store. This means that all M files are of the same (unit) size, placing a constraint on the cardinality of cache contents $|\mathcal{B}_n| \leq K$. The important size parameters of the system are N, M, K. The interesting regime that we will study is

$$K < M \leq KN. \tag{1}$$

The first inequality implies that the cache size K is not enough to fit all files and hence each node needs to make a selection of files to cache. The second inequality requires that the total network cache capacity KN (summing up all individual caches) is sufficient to store all files at least once, and thus each content request can be served.

B. Multihop Capacity Optimization: Relaxation & Rounding

Due to placement constraints $|\mathcal{B}_n| \leq K$, the structure of cache optimization problems is inherently combinatorial, and thus they are hard to solve. In this section (i) we state the relevant problems for optimizing capacity scaling laws with caching, (ii) propose a relaxed problem based on file densities which is easy to solve, and in this way (iii) establish a feasible solution which is of the same order with the original combinatorial problem.

[1]Note that the grid model is not accurate for wireline networks where the diameter scales as $\log N$, as node connectivity follows power laws [11].

1) Joint Replication-Routing Problem: To yield the correct scaling law of required link capacities, it is necessary to select the best placement and the best routing that minimizes the load at the worst link in the network. Let C_ℓ be the traffic load carried by link ℓ. The network is stable, not discarding requests, only if the capacity of link ℓ exceeds C_ℓ for all ℓ. The link loads $[C_\ell]$ can be adjusted by (i) caching popular content (by means of choosing the placement $[\mathcal{B}_n]$) and (ii) load balancing across a set of routing paths $[\mathcal{R}_{n,k}]$ from caching node n to requesting node k. Hence the correct scaling law of the required link capacities C^* is identified as the value of the following optimization:

PROBLEM 1: $C^* =$ *Minimize*$_{[\mathcal{B}_n],[\mathcal{R}_{n,m}]} \max_\ell C_\ell$, *s.t.*

1) $|\mathcal{B}_n| \leq K, \quad \forall n$ *(cache constraint),*
2) $\sum_n 1_{\{m \in \mathcal{B}_n\}} \geq 1, \quad \forall m \in \mathcal{M}$ *(all files are cached),*
3) $[\mathcal{R}_{n,m}]$ *are feasible routes*

This joint optimization turns out to be a hard combinatorial problem[2], not amenable to an easy-to-compute solution. Therefore, we resort to simplifications and approximations that provide an order-optimal solution, i.e., whose value of the objective function is within a constant to the optimal, hard-to-compute $\min \max C_\ell$. A first step that preserves the order-optimality of the solution is the relaxation of the objective to the average link traffic $\text{avg}_\ell C_\ell$; then routing variables $[\mathcal{R}_{n,k}]$ can be fixed to shortest paths without loosing optimality; these steps are explained in detail in [8]. However, the decisive step involves breaking the coupling between the individual caches $[\mathcal{B}_n]$. To achieve this decoupling, we introduce a new notion, that of *replication density* of content m.

2) Relaxed Density-based Problem: Given a placement $[\mathcal{B}_n]$, consider the frequency of occurrence of each file m in the caches, or *replication density* d_m as the fraction of nodes that store file m in the network:

$$d_m = \frac{1}{N} \sum_{n \in \mathcal{N}} 1_{\{m \in \mathcal{B}_n\}}.$$

Based on this metric, we define a simpler problem:

PROBLEM 2: $C =$ *Minimize*$_{[d_m]} \sum_{m \in \mathcal{M}} \left(\frac{1}{\sqrt{d_m}} - 1 \right) p_m$, *s.t.*

1) For any $m \in \mathcal{M}$, $1/N \leq d_m \leq 1$,
2) $\sum_{m \in \mathcal{M}} d_m \leq K$.

In the above, we optimize on the densities d_m, which express the fraction of caches containing file m. In the objective, $1/\sqrt{d_m} - 1$ approximates (in-order) the average hop count from a random node to a cache containing m. Weighted by the probability p_m of requests on m, the summation expresses the average link load per request. Additionally, the constraint $\sum_{m \in \mathcal{M}} d_m \leq K$ reflects another relaxation, whereby the cache size constraint is only satisfied on the average in the network. Observe that we have removed the individual cache constraints, and a feasible solution to problem 2 may yield file densities that correspond to caching more than K files in one node.

It is clear that any feasible solution of problem 1 yields a file density $[d_m]$ that is feasible for problem 2, but not vice versa, hence the density-based formulation is a relaxed version of the original min-max problem [8] and we have

$$C = O(C^*).$$

In particular, problem 2 is convex and its unique solution can be found using the Karush-Kuhn-Tucker (KKT) conditions. Regarding the constraints on d_m about its minimum and maximum value, either one of them can be an equality, or none. This partitions $\mathcal{M}$ into three subsets, the 'up-truncated' $\mathcal{M}_\uparrow = \{m : d_m = 1\}$ containing files stored at all nodes, the 'down-truncated' $\mathcal{M}_\downarrow = \{m : d_m = 1/N\}$ containing files stored in just one node, and the complementary 'non-truncated' $\mathcal{M}_\updownarrow = \mathcal{M} \setminus (\mathcal{M}_\uparrow \cup \mathcal{M}_\downarrow)$ of files with $1/N < d_m < 1$. Arranging p_m in decreasing order, the partitions become $\mathcal{M}_\uparrow = \{1, 2, \ldots, l-1\}$, $\mathcal{M}_\updownarrow = \{l, l+1, \ldots, r-1\}$, and $\mathcal{M}_\downarrow = \{r, r+2, \ldots, M\}$; l and r are integers with $1 \leq l \leq r \leq M+1$. The solution d_m is equal to

[2]The cache optimization problem of interest bears resemblance to *database location problem*, see [15].

$$d_m = \begin{cases} 1, & m \in \mathcal{M}_{\uparrow}, \quad (2a) \\ \dfrac{K - l + 1 - \frac{M-r+1}{N}}{\sum_{j \in \mathcal{M}_{\updownarrow}} p_j^{\frac{2}{3}}} \, p_m^{\frac{2}{3}}, & m \in \mathcal{M}_{\updownarrow}, \quad (2b) \\ 1/N, & m \in \mathcal{M}_{\downarrow}. \quad (2c) \end{cases}$$

Fig. 2 illustrates such an example solution, depicting the density d_m, indices l and r, as well as sets $\mathcal{M}_{\uparrow}$, $\mathcal{M}_{\updownarrow}$, $\mathcal{M}_{\downarrow}$ when file popularities follow the Zipf law (see Section II-C).

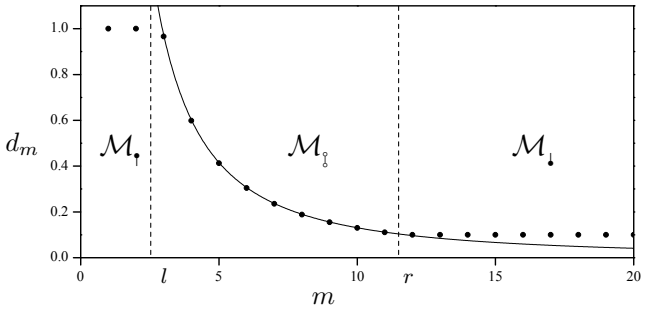

Fig. 2. An example case of density d_m and the $\mathcal{M}_{\uparrow}, \mathcal{M}_{\updownarrow}$ and $\mathcal{M}_{\downarrow}$ partitions. Solid line plots the $\sim m^{-\frac{2\tau}{3}}$ law of $m \in \mathcal{M}_{\updownarrow}$, when p_m follows Zipf's law.

3) Discrete Density (Rounding): The solution (2) is not directly mapped to a feasible solution for Problem 1, but we may construct one using a two-step process, (i) first rounding $[d_m]$ to $[d_m^\circ]$, and (ii) second placing the content symmetrically on the network according to $[d_m^\circ]$ so that the constraints $|\mathcal{B}_n| \leq K$ are satisfied. For (i), we simply define $d_m^\circ \triangleq 4^{-\nu_m^\circ}$ rounded to the largest power less or equal to $[d_m]$

$$d_m^\circ \triangleq \max\left\{4^{-i} \colon 4^{-i} \leq d_m, \ i \in \{0, 1, \ldots, \nu\}\right\}. \quad (3)$$

Then for (ii), [8] gives an algorithm to allocate the files $\mathcal{M}$ in the caches $[\mathcal{B}_n]$ given the replication densities d_m°. The algorithm can be explained by means of Figure 3. We begin with the grey file which has $d_0^\circ = 1$, this file is simply cached everywhere. Then for the file with $d_0^\circ = 1/4$, we focus on a 2×2 subgrid of nodes (any such subgrid suffices but it helps to fix an origin as a reference). In this subgrid we try to fill the diagonal, which in this case is achieved by placing file 1 at the coordinate $(1, 1)$ (top left node in the grid). As a last step for this file, we place replicas by tiling the subgrid everywhere in the network. The result is that file 1 is replicated with density 1/4, as prescribed by the solution. Then for the files with density 1/16 we enlarge the subgrid to 4×4. In general the subgrid has a size $2^{\nu_m^\circ} \times 2^{\nu_m^\circ}$ and is aligned with all the considered subgrids. We then fill the subgrid with the new files starting with the diagonal, notably files 2, and 3 in the example. Then files 4, 5, 6 are filled in the second diagonal which is below the first, while 7 completes the second diagonal by wrapping up at the coordinate $(1, 4)$. During the filling, we only select nodes that have less files than the maximum. For example, when we are filling object 2 in the subgrid 4×4, we skip the node $(1, 1)$ since that node has already two files (file 0 and file 1), and we place file 2 at node $(2, 2)$ which only had one file so far (file 0). Where would we place file 10 with $d_{10}^\circ = 1/64$? We would consider the subgrid 8×8 (i.e. the entire grid), the first three diagonals are fully filled, hence we would place it in the first open spot in the fourth diagonal, that is node $(4, 1)$.

Using this placement, we ultimately obtain a feasible solution for Problem 1 with value $C^\circ = \Omega(C)$. Finally, the following result is established in [8].

THEOREM 1 [RELAXATION & ROUNDING YIELDS ORDER OPTIMAL LINK LOADS]: *There exist positive constants a, b that depend on the distribution $[p_m]$, and cache capacity K, such that*

$$C^* \leq C^\circ \leq aC^* + b.$$

Furthermore it is $\Theta(C^*) = \Theta(C^\circ) = \Theta(C)$.

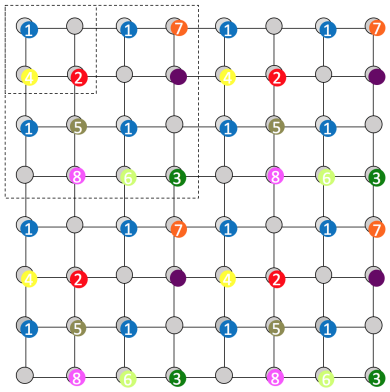

Fig. 3. An example of the canonical rounding used to obtain a feasible cache placement solution. Here, the following items are placed $d_0^\circ = 1$, $d_1^\circ = 1/4$, $d_2^\circ = \cdots = d_8^\circ = 1/16$.

In conclusion, we may derive the scaling laws of wireless networks with caching C^* by directly using the relaxed Problem 2 solution, C. We remark that since a, b above depend on K, scaling laws with respect to K need to consider this dependence carefully. This is taken into account in the analysis of [10].

C. Asymptotic Laws for Zipf Popularity

To study the scaling of C, we switch from the arbitrary popularity to the Zipf law, which has been observed to model well the content popularity of the traffic of WWW and other types of services in numerous traces in the literature [12], [14], [16]–[20]. It is defined as follows:

$$p_m = \frac{1}{H_\tau(M)}\, m^{-\tau}, \tag{4}$$

where $H_\tau(n) \triangleq \sum_{j=1}^{n} j^{-\tau}$ is the truncated zeta function evaluated at τ (also called the n^{th} τ*-order generalized harmonic number*), and τ is the parameter of the distribution, adjusting the rate of popularity decline with m. Regarding parameter τ, values ≤ 1 are reported for web traffic, [21]–[23]. Interestingly, [24] measures $\tau = 1$ at proxies versus 1.4–1.6 at a 'busy' web server. Analyzing traces from mobile browsing, [25] report values in the range of 1–1.5, while in User Generated Content (UGC) video, [26] fits popularity with a combination of Zipf and Exponential Cut-off, where the Zipf component has τ in the ranges of about 0.98–1.47 and 0.45–1.09 in two different scenarios. Last, content popularity in P2P systems was fitted with a $\tau = 0.95$ in [27], and in a Video on Demand system with 0.70 in [28]. As there is no conclusive answer about the value of τ, we consider all possibilities in our investigation.

We derive an approximation for $H_\tau(n)$ by bounding the sum: for $n \geq m \geq 0$,

$$\begin{aligned} &\int_m^n (x+1)^{-\tau} dx \leq H_\tau(n) - H_\tau(m) \leq 1 + \int_{m+1}^n x^{-\tau} dx, \Rightarrow \\ &\begin{cases} \frac{(n+1)^{1-\tau} - (m+1)^{1-\tau}}{1-\tau} \leq H_\tau(n) - H_\tau(m) \leq \frac{n^{1-\tau} - (m+1)^{1-\tau}}{1-\tau} + 1, & \text{if } \tau \neq 1, \\ \ln \frac{n+1}{m+1} \leq H_\tau(n) - H_\tau(m) \leq \ln \frac{n+1}{m+2}, & \text{if } \tau = 1. \end{cases} \end{aligned} \tag{5}$$

Substituting the solution (2) and plugging in the Zipf distribution into the objective of Problem 2, it follows that

$$C \triangleq \sum_{m \in \mathcal{M}} \left(d_m^{-\frac{1}{2}} - 1 \right) p_m = C_{\text{i}} + C_{\text{ii}} - \sum_{j=l}^{M} p_m, \tag{6}$$

where $\sum_{j=l}^{M} p_m = O(1)$ (as it lies always in $[0,1]$), and

$$C_{\text{i}} \triangleq \sum_{m \in \mathcal{M}_{\text{i}}} \frac{p_m}{\sqrt{d_m}} \overset{(2),(4)}{=} \frac{\left[H_{\frac{2\tau}{3}}(r-1) - H_{\frac{2\tau}{3}}(l-1) \right]^{\frac{3}{2}}}{\sqrt{K - l + 1 - \frac{M-r+1}{N}}\; H_\tau(M)}, \tag{7}$$

TABLE I
SCALING LAWS WITH CONSTANT CACHE SIZE

M		M finite	$N\to\infty$, then $M\to\infty$	$M\sim KN$, hence $M=\Theta(N)$	
				$KN-M=\omega(1)$	$KN-M=O(1)$
C	$\tau<1$	$\Theta(1)$	$\Theta\left(\sqrt{M}\right)$	$\Theta\left(\sqrt{N}\right)$	$\Theta\left(\sqrt{N}\right)$
	$1<\tau<\frac{3}{2}$	$\Theta(1)$	$\Theta\left(M^{\frac{3}{2}-\tau}\right)$	$\Theta\left(\frac{\sqrt{N}}{(KN-M)^{\tau-1}}\right)$	$\Theta\left(\sqrt{N}\right)$
	$\tau>\frac{3}{2}$	$\Theta(1)$	$\Theta(1)$	$\Theta\left(\frac{\sqrt{N}}{(KN-M)^{\frac{3(\tau-1)}{2\tau}}}\right)$	$\Theta\left(\sqrt{N}\right)$

$$C_{\downarrow} \triangleq \sum_{m\in\mathcal{M}_{\downarrow}} \frac{p_m}{\sqrt{d_m}} \overset{(2),(4)}{=} \sqrt{N}\ \frac{H_\tau(M)-H_\tau(r-1)}{H_\tau(M)}. \tag{8}$$

To analytically compute the law of C, we need to analyze l and r. Note that since M scales to infinity, l, r may also scale to infinity, or not, depending on the actual solution (2). Moreover, observing that the expressions (7)-(8) depend on $H_\tau, H_{\frac{2\tau}{3}}$, we expect different cases to appear due to the form of (5). In fact, previous work [8] yields 5 cases depending on the values of τ. These in terms are mapped to scaling laws via (7)-(8). The results are presented in Table I. The cases $\tau=1$ and $\tau=3/2$ are omitted to avoid clutter; they are similar to the cases $\tau<1$ and $\tau>3/2$ up to a logarithmic factor.

D. Scaling Laws for Constant Cache Size K

Table I shows how the solution of Problem 2 scales with the system size parameters N (number of nodes/users), and M (catalog size); K (cache size) in this table is a constant. From Theorem 1, the scaling of C also applies to the required link capacity for sustaining a uniform request rate $\lambda=1$. For example, a scaling $\Theta(1)$ means that the network can sustain unit throughput even if the link capacities are (sufficiently large) constants; this is the desirable case. On the other hand $\Theta(\sqrt{M})$ means that to sustain unit throughput, the link capacity needs to increase proportionally to the square root of the file catalog. Since in practical systems M is expected to grow with the network size N, while the link capacity is expected to obey certain limits (like the Shannon limit), this means that the network will be unsustainable under such a law. Recall that we may also inverse the scaling to see how throughput will scale in a network with fixed link capacities. For example if we invert the law $\Theta(\sqrt{N})$ it gives $\lambda=\Theta(1/\sqrt{N})$, which is the Gupta-Kumar scaling for random communicating pairs in a scaling network [2]. This law appears in our analysis whenever the replication capacity $KN-M=\Theta(1)$ is low, i.e. almost all cache slots are used to store each file once–see the last column of table I.

Since K is constant, the results are derived for different cases of how N and M scale to infinity together. The second column corresponds to the case where only N scales and M is fixed, in which case the system is sustainable (required link capacity $\Theta(1)$). However, in practical systems the file catalog grows with the user population.

The third column refers to the case where we take the limits in order, first $N\to\infty$ and then $M\to\infty$, i.e. we have $M=o(N)$ and the catalog grows sublinearly to the network size. The scaling laws in this case relate to the file catalog size, which is a slight improvement over the Gupta-Kumar scaling. Different scaling is obtained for different values of τ, where larger values of τ improve the sustainability (decrease the required link capacity).

In the last two columns we set $M=\alpha KN$ for some positive $\alpha<1$, and then take the limit $N\to\infty$. The scaling laws in this case also depend on $KN-M$, i.e. how many replication slots are left in the caches after we store each file once. There are two cases, (a) $KN-M=\omega(1)$, and (b) $KN-M=O(1)$. In the first case, despite $M=\alpha KN$, there are enough replication slots to yield scalings that represent improvements over the Gupta-Kumar scaling as long as $\tau>1$. In the second case, we obtain the Gupta-Kumar scaling irrespective of the value of τ.

TABLE II
SCALING LAWS WITH SCALING CACHE SIZE

		'High' K	'Low' K
C	$\tau < 1$	$O\left(\sqrt{\frac{M}{K}}\right)$	$\Theta\left(\sqrt{N}\right)$
	$1 < \tau < \frac{3}{2}$	$O\left(\frac{M^{\frac{3}{2}-\tau}}{\sqrt{K}}\right)$	$\Theta\left(\frac{\sqrt{N}}{(KN-M)^{\tau-1}}\right)$
	$\tau > \frac{3}{2}$	$O(1)$	$\Theta\left(\sqrt{\frac{N}{KN-M}}\right)$

E. Scaling the Cache Size $K \to \infty$

As memory becomes cheaper and cheaper, we may envisage the scenario where the cache size per node K also scales to infinity. The extention of the analysis of (7)-(8) in the case $K \to \infty$ is found in [10]. Here we briefly discuss the resulting scaling laws presented in table II.

Considering how N, M, K grow, we can study the system in different regimes of operation, which complicates the exposition of scaling laws. For this reason, we focus here on two specific regimes of interest. We compare the total network memory KN to the file catalog M, and split the analysis to two cases (i) $KN = \Omega(M)$ and (ii) $KN = O(M)$ (in fact [10] provides also the asymptotic constant that separates the two regimes. The first case is called 'High' K and the second 'Low' K.

1) 'High' K: The most interesting regime to explore for perfect sustainability is the one of $C = O(1)$. As the formulas show, to keep C bounded, the hardest case is on $\tau < 1$: node cache capacity K should scale as fast as content volume M. In the intermediate case of $1 < \tau < 3/2$, node capacity K has to scale with M, but slower, at a sublinear power. The case of $\tau > 3/2$ is quite interesting, as $C = O(1)$ always holds true.

2) 'Low' K: This regime is characterized as unsustainable, because the increase of network node cache K and/or node count N against content volume M is not sufficient to keep C low. First, note that when the replication storage capacity beyond the storage of the primary copy is limited, i.e., $KN - M = O(1)$, C scales as fast as $\sqrt{N}$, the law of [2]. Observe, moreover, that for $\tau > 1$, $\sqrt{N}$ is scaled down by the capacity $KN - M$ available for replication beyond the primary copy at some power equal to$\sqrt{KN - M}$ or $(KN - M)^{\tau-1}$ for $\tau > 3/2$ or $1 < \tau < 3/2$, respectively; this quantifies the gain from adding extra capacity in the network beyond the one required to store a primary copy per file.

F. Discussion about the sustainability of wireless caching networks

We presented a joint delivery and replication problem that leads to the characterization of the sustainability of multihop wireless networks with caching. After formulating the precise combinatorial problem on square lattice networks, we relaxed it to a simple density-based problem, whose analysis permitted to derive the scaling laws of the required link capacity. The key factors in network sustainability are a) the file popularity power law parameter τ, as well as b) the relative scaling of the numbers of files vs. network nodes.

While our treatment so far characterizes the caching benefits for multihop networks, a different line of work examines the caching scaling laws in the wireless broadcast medium using coding, [29], [30]. Next, we continue our exposition zooming in the cellular topology of wireless networks, considering the exploitation of caching in small cells with device-device cooperation.

III. CACHE-ENABLED SMALL CELL NETWORKS: MODELLING AND DESING TRADEOFFS

Cache-enabled small cell networks (SCNs) have been proposed as a potential solution for alleviating the backhaul bottleneck problem [31]–[33]. The main idea is to introduce cache capacity at small cell base stations (SBSs) to prefetch popular contents during off-peak hours before being requested by local users. Caching in wireless networks also exploits the high degree of asynchronous content reuse caused by information-centric applications such as video-on-demand (VoD), social networks and content sharing. When users request for some popular contents already cached in the local SBSs, the service latency is largely reduced since it does not need to pass through the

backhaul to retrieve the content from remote servers. The improved energy efficiency is also an important benefit of small cell caching mainly due to the fact that repeated transmissions of the same content from the core network to local SBSs are avoided.

A. Cache-enabled SCNs with Local User Interest Correlation

Prior studies investigating wireless caching problems often assume similar popularity pattern for all users in the system. Under the consideration of different interest levels over requested content, a clustering algorithm is proposed in [33] to group users into clusters based on their request profile. Optimal caching decisions are then made at each SBS by using a distributed regret learning approach to minimize the total service delay.

We are interested here in the case where caching decisions are made at SBSs based on local users' request pattern rather than the global content popularity, since in cache-enabled SCNs each SBS only serves a small amount of users. Specifically, each SBS constructs its local regular content library by sampling regularly requested files of users in its sampling range, then determines the set of files to be cached under limited cache capacity. We choose the conventional "cache the most popular files" strategy to give baseline analysis on the impact of user interest correlation on the cache hit/miss probability. The results can be generalized to cases with more advanced caching strategies as proposed in [34]. We assume Zipf-like distribution for the local regular content popularity in the same region. Under stochastic spatial models for the SBS locations and user distribution, we derive the cache service probability as a function of the maximum sampling distance. We also consider that there is a sampling cost, which is modelled as a distance-dependent function. The tradeoff between the cache service probability and the sampling cost is also studied, and the optimal sampling distance subject to a cost constraint is provided.

1) Network Model and Analysis: We consider a cache-enabled SCN model with limited storage space at each SBS, serving its nearby users within a certain distance according to its transmit power constraint. We model the SBS distribution on the two-dimensional Euclidean plane $\mathbb{R}^2$ by a homogeneous PPP $\Phi_c = \{Y_i, i \in \mathbb{N}^+\}$ with intensity λ_c, where Y_i denotes the position of the i-th SBS. Users to be served are distributed according to another independent homogeneous PPP $\Phi_u = \{x_j, j \in \mathbb{N}^+\}$ with intensity λ_u, where x_j denotes the position of the j-th user.

We introduce the notion of "local user interest pattern", which can be acquired when allowing each SBS to sample the regularly requested content of all users in its sampling region of radius R_p. The small cell cachers decide which files to cache based on the adopted caching policy. Assume there exists a maximum distance that a SBS can serve, denoted by R_v, the disk centered at the SBS with radius R_v can be seen as the service region of this SBS.

When increasing the sampling range, the SBSs may have better knowledge of local user interest pattern, but due to the limited cache storage only the most popular files will be stored. Then, for a random user being sampled the probability of finding its regular content cached at its covering SBSs will decrease with the sampling region size. The service range constraint also gives un upper bound on the probability of an arbitrary user being served by the small cell cachers. We denote this service probability by P_{sv}. Increasing the sampling range also leads to higher cost of the local request pattern learning at the SBSs. In this paper we focus on the influence of the sampling range on the service probability and the sampling cost for different levels of local interest correlation among small cell users.

Suppose that each user has a library of size J, which contains its regularly requested files. For simplicity, we assume that all the files have equal unit size. Each SBS has limited cache storage size denoted by M, which is the maximum number of files that can be stored. Denoting by N the number of users in the sampling region of a SBS, from point process theory, we have $\mathbb{E}[N] = \lambda_u \pi R_p^2$ [35].

When users in the same sampling region share similar interest in requested content, different users may have overlapping files in their regular content libraries, so the overall regular content library size will not grow linearly with the number of users in this area. Conditioning on having N users in the sampling region of a typical SBS, we denote by $\mathcal{C} = \{c_1, \ldots, c_S\}$ the local content library of the N users with S representing the library size. Denote by $g : \mathbb{N}^+ \to \mathbb{N}^+$ the mapping function from the number of users N to the local library size S. In reality $S = g(N)$ can be learned numerically at the SBSs and the approximate function can be found by data fitting. Here we assume that g is a piecewise function and follows $S = \min\left(JN, \lceil J\left(1 + \mu \log N\right)\rceil\right)$, where μ is a constant factor that characterizes the similarity level of local user interest.

For a given local regular content library, we assume that the popularity distribution of files in it follows the Zipf-Mandelbrot law, that is, for the i-th most popular file we have its request probability as

$$p_i = \frac{\Omega}{(q+i)^\gamma} \tag{9}$$

where $\Omega = \left(\sum_{n=1}^{S} (q+n)^\gamma \right)^{-1}$ is the normalization constant, q is the shift parameter which is related to the shifting based on the Zipf distribution, and γ is the shape parameter, which defines the concentration (correlation) level of the content popularity. Similarly to the g function, the real popularity distribution can be learned by sampling and sorting the popularity of all requested files in the local region.

Note that μ in the g function defines how similar the regular content libraries of users in the same area are, γ in the popularity distribution defines the relative disparity of content popularity in the local library. The combination of these two factors gives full characterization of the correlation level of local user interest.

Service Probability Analysis. We consider at first a baseline caching policy, which is "cache the most popular files". Cache entities at the SBSs have identical and limited storage size as M files. Assume that we have a SBS at the origin with N users in its sampling region of radius R_p. The local content library size is known as S. When $M \geq S$, all files in the local content library can be stored at the SBS. When $M < S$, due to the limited cache storage only the most popular M files in the sampling region can be cached. The cache miss probability P_m, which is the probability that a random file in the local regular content library $\mathcal{C}$ not being cached, can be given as

$$P_m = \sum_{i=M+1}^{S} p_i = \sum_{i=M+1}^{S} \frac{1}{(q+i)^\gamma \sum_{n=1}^{S} (q+n)^\gamma}. \tag{10}$$

Evidently, P_m increases with S, which verifies the intuition that larger content library size leads to higher cache miss probability when the cache storage capacity is limited. Assume that users outside the sampling range of a SBS have almost negligible probability to find their regularly requested content in this small cell cache. The service probability P_{sv} is then denoted by the probability of the intersection of three events:

- $\mathcal{E}_1 =$ user's regular content library being sampled by at least one SBS,
- $\mathcal{E}_2 =$ the requested content being cached,
- $\mathcal{E}_3 =$ user in the service range of the sampling SBS.

Furthermore we have $\mathcal{E}_1 \subseteq \mathcal{E}_3$ if $R_p \leq R_v$ and $\mathcal{E}_3 \subset \mathcal{E}_1$ otherwise. In the case where $R_p \leq R_v$, P_{sv} can be given by

$$\begin{aligned} P_{sv} &= \mathbb{P}[\text{user being sampled} \cap \text{content being cached}] \\ &= 1 - \mathbb{P}[\text{not being sampled}] \\ &\quad - \sum_{k=1}^{\infty} \mathbb{P}[\text{sampled by k cachers}] \cdot \mathbb{P}[\text{k cache miss}] \\ &= 1 - e^{-\lambda_c \pi R_p^2} \\ &\quad - \sum_{k=1}^{\infty} \frac{\left(\lambda_c \pi R_p^2\right)^k}{k!} \exp\left(-\lambda_c \pi R_p^2\right) \prod_{j=1}^{k} P_m^j, \end{aligned} \tag{11}$$

where P_m^j is the cache miss probability of the j-th SBS which has the typical user within its sampling range. When different small cells have different request correlation levels, which corresponds to different γ in the Zipf distribution, the cache miss probability can be given as

$$P_m^j = \sum_{i=M+1}^{S} \frac{1}{(q+i)^{\gamma_j} \sum_{n=1}^{S} (q+n)^{\gamma_j}}, \tag{12}$$

where γ_j is the shape parameter of the content popularity distribution in the sampling region of the j-th SBS. It is easy to see that P_{sv} increases monotonically with R_p, meaning that larger sampling distance gives higher

probability of a random user to find its regularly requested content cached in the covering SBSs.

In the case where $R_p > R_v$, the service probability is given by

$$\begin{aligned} P_{sv} &= \mathbb{P}[\text{user in service range} \cap \text{content being cached}] \\ &= 1 - e^{-\lambda_c \pi R_v^2} \\ &\quad - \sum_{k=1}^{\infty} \frac{\left(\lambda_c \pi R_v^2\right)^k}{k!} \exp\left(-\lambda_c \pi R_v^2\right) \prod_{j=1}^{k} P_m^j. \end{aligned} \tag{13}$$

Contrary to the previous case, for fixed R_v the service probability decreases with R_p, which is easy to understand since larger sampling region corresponds to higher cache miss probability for users being sampled, while the number of users that can be served will not increase with the sampling range due to the service range constraint. Combining these two cases we have

$$\begin{aligned} P_{sv} &= \mathbb{P}[\text{user in service range} \cap \text{content being cached}] \\ &= 1 - e^{-\lambda_c \pi \min\{R_p, R_v\}^2} - \sum_{k=1}^{\infty} e^{-\lambda_c \pi \min\{R_p, R_v\}^2} \cdot \\ &\quad \frac{\left(\lambda_c \pi \min\{R_p, R_v\}^2\right)^k}{k!} \prod_{j=1}^{k} P_m^j. \end{aligned} \tag{14}$$

From the above result we see that for a given constraint on the service range of a SBS, the service probability increases with R_p till it reaches the service range limit R_v, then it decreases monotonically with R_p.

Sampling Cost Analysis The regular content sampling allows the SBSs to learn the interests of local users, thus the service offered by small cell cachers can be better adapted to the local request pattern. On the other hand, the power consumption for the sampling procedure must be taken into consideration for the purpose of energy efficiency. Denote by C_s the sampling cost of the typical SBS at the origin with sampling region radius R_p, we introduce a cost function for the sampling process, which depends on the distance from the sampled users to the SBS, i.e.

$$C_s = \sum_{j \in \Phi_u \bigcap \mathcal{B}(0,R_p)} f(\|x_j - Y_0\|) \cdot \eta \tag{15}$$

where $\|x_j - Y_0\|$ is the Euclidean distance between the j-th user in the sampling region and the typical SBS at the origin, f is a distance-dependent cost function, η is a constant cost factor. For simplicity we choose $\eta = 1$ and inverse pathloss model for f as $f(x) = x^{\alpha}$ where α is the pathloss exponent.

From Campbell's theorem, we have the mean cost when averaging over different realizations of PPP as

$$\begin{aligned} \mathbb{E}[C_s] &= \mathbb{E}\left[\sum_{j \in \Phi_u \bigcap \mathcal{B}(0,R_p)} \|x_j - Y_0\|^{\alpha}\right] \\ &= 2\pi\lambda_u \int_0^{R_p} r^{\alpha} \cdot r \mathrm{d}r \\ &= \frac{2\pi\lambda_u}{\alpha+2}\left(R_p^{\alpha+2} - 1\right). \end{aligned} \tag{16}$$

We see that the sampling cost increases linearly with the user density and follows power law of order $\alpha + 2$ growth with the sampling region radius. Note that in dense network deployments, larger sampling distance results in improved learning of the request pattern; however the sampling cost may be prohibitively high and should be taken into account when designing the optimal sampling distance.

2) Optimal Sampling Range under Cost Constraint: In Section III-A1 we gave basic results on the service probability and sampling cost for cache-enabled small cell networks with local request pattern learning. Based on our assumptions on user interest correlation, we can gain insight on how the service probability and cost scale with the sampling range.

Fig. 4 shows an example of the service probability as a function of the sampling region radius when there are

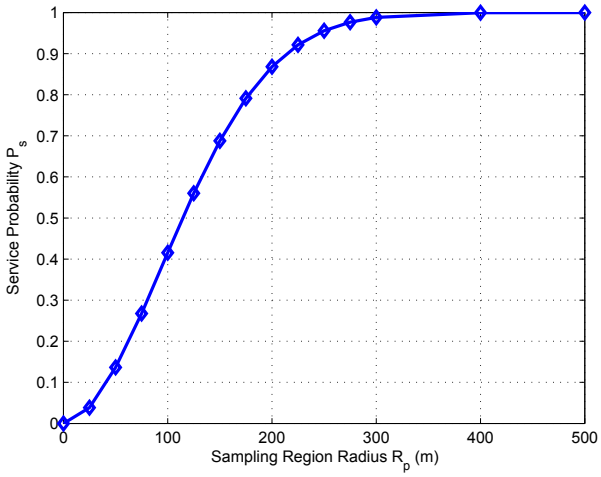

Fig. 4. Service probability vs. sampling range with infinite service range. $q = 0$, $\gamma = 1$.

no constraint on the service range. We see that increasing sampling range gives higher service probability, but the increment tends to saturate when R_v is relatively large. By taking into consideration the limited service range while ignoring the sampling cost, the optimal sampling region size is equal to R_v since it gives the highest service probability, as we have analyzed in Section III-A1. In this section, we search the optimal sampling range which gives sufficiently good service probability while assuring the sampling cost is below a given constraint.

We formulate the problem as follow:

$$R_p^* = \arg\max_{R_p} P_{sv} \tag{17}$$

Subject to:

$$R_p \in [0, R_v], \tag{18}$$

$$C_s(R_p^*) < C_{\max}, \tag{19}$$

$C_{\max}$ is the predefined cost constraint. The first condition comes from the fact that when $R_p > R_v$ the service probability decreases monotonically with R_p. The second condition is due to the sampling cost constraint.

With the help of (16), we can derive the constraint on R_p to satisfy (19) as

$$R_p^* < \left(\frac{C_{\max}(\alpha+2)}{2\pi\lambda_u} + 1 \right)^{\frac{1}{\alpha+2}}. \tag{20}$$

Since P_{sv} is a monotonically increasing function of R_p for $R_p \in [0, R_v]$, combining these two constraints we have the optimal sampling distance given by $R_p^* = \min\left(R_v, \left(\frac{C_{\max}(\alpha+2)}{2\pi\lambda_u} + 1\right)^{\frac{1}{\alpha+2}}\right)$.

3) Numerical Results: In this section we validate our analysis on cache-enabled small cell networks with local user interest sampling. We show the tradeoff between the small cell cache service probability and the sampling cost for different levels of local user interest correlation and user densities.

We set $\lambda_c = 2 \times 10^{-5}$ and $\lambda_u = 8 \times 10^{-4}$ as the density of SBSs and of users to be served. The service region of each SBS has radius $R_v = 200$ m. The regular content library size of each user is $J = 10$. The maximum cache storage of a SBS is $M = 50$. We use $S = \min\left(JN, \lceil J\left(1 + 3\log N\right)\rceil\right)$ as the mapping function from the number of users to the overall local content library size. The popularity distribution is assumed to follow the Zipf-Mandelbrot law with $q = 0$ (Zipf's law). We assume the same shape parameter for the content popularity distribution in the sampling region of all the SBSs, meaning that $\gamma_j = \gamma$ for $j = 1, \cdots, \infty$. Note that considering uniform or different shape parameters for the popularity distribution does not change the general trend of how the sampling distance affects the service probability P_{sv}. Numerical results are presented with different values of the shape parameter $\gamma \in [0.25, 1.5]$, which correspond to different user interest correlation levels, in order to see how the cache service

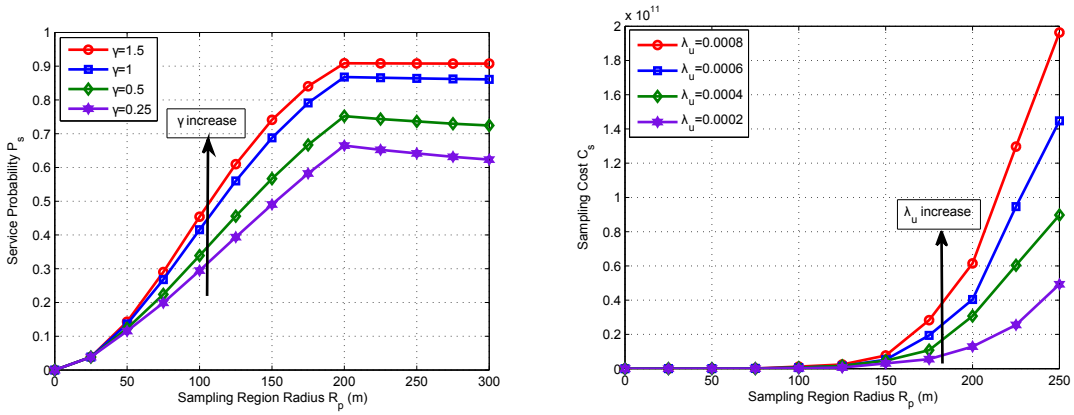

Fig. 5. (left) Service probability under different levels of user interest correlation. (right) Sampling cost vs. sampling range with distance-dependent cost function.

performance scales with local user interest pattern.

B. Service Probability under Different Correlation Levels of Users' Interest

In Section III-A1, we saw that the cache service performance depends on the local user interest correlation level. In Fig. 5-(left) we plot the service probability P_{sv} as a function of the sampling range R_p for $\gamma = 0.25$, 0.5, 1 and 1.5 respectively.

From Fig. 5-(left), we see that P_{sv} increases with R_p before it reaches $R_v = 200$ m then decreases when R_p continues to grow. However, the speed of decrease is relatively small compared to the growth in $R_p \in [0, R_v]$, especially for higher values of γ. This is because higher γ corresponds to higher concentration of content popularity. Though increasing library size increases the number of files not being cached, the most popular files are cached with higher probability, which still leads to lower cache miss probability in the sampling region.

Sampling Cost. Fig. 5-(right) shows the simulated sampling cost per SBS as a function of the maximum sampling distance with the adopted distance-dependent cost function. The simulation results are obtained under different user density settings and by averaging over 5000 PPP realizations. It shows the dramatic growth of the cost of local regular content sampling when increasing the sampling distance R_p. In a high user density scenario, the cost issue is of growing importance, showing the need for taking into consideration the sampling cost when searching for the optimal sampling distance.

1) Conclusions: We investigated cache-enabled SCNs with local user interest sampling managed by SBSs for local request pattern learning. We provided analytical and numerical results on the cache service performance and the sampling cost, as well as the optimal sampling distance to maximize the cache service probability under a given cost constraint. The main takeway of this paper is the idea of local regular content sampling and the influence of the sampling distance on the cache service performance, which will be beneficial for designing the sampling procedure of cache-enabled SCNs.

IV. D2D Caching vs. Small Cell Caching

Proximity-based content caching and distribution in wireless networks has been identified as a promising traffic offloading solution for improving the capacity and the quality of experience (QoE) by exploiting content popularity and spatio-temporal request correlation.

The concept of caching at the radio access network edge, termed here as small cell (SC) caching, is driven by the bandwidth and latency gains from introducing caching capabilities to the small base stations (SBSs) and reducing the backhaul utilization. In [32], the concept of FemtoCaching with distributed cache helpers is proposed and the optimum way of assigning files to cache helpers is analyzed by minimizing the expected downloading time. [31] studies the performance of cache-enabled small cell networks, deriving closed-form expressions for the outage probability and average delivery rate under stochastic spatial distribution of the SBSs. A different approach is to exploit the available storage space of mobile devices and distribute cached content to other devices in proximity

through D2D communication. [36] discusses the throughput-outage tradeoff of D2D caching networks using a protocol model for the spatial scheduling of coexisting D2D links. In [37] authors study the scaling behaviour of a D2D network and give a closed-form expression for the optimal collaboration distance as a function of the system model parameters. [38] considers a random caching strategy for stochastically distributed caching-server devices to serve nearby user devices and proposes an algorithm to find the optimal caching probabilities of contents in the request library.

At first sight, D2D caching and SC caching may bring comparable gains in terms of network throughput, area spectral efficiency and energy efficiency. However, they are quite different in the following aspects:

- **Cache capacity**. Caching entities deployed in SBSs can have very large storage capability thanks to the low cost of storage units. In contrast, user devices, such as cellphones and tablets, have relatively small storage capacity and can only serve a rather small amount of requests generated by devices in their proximity.
- **Transmit power and coverage**. User devices normally transmit with much less power than SBSs, which in turn corresponds to smaller covering range. As a result, the success probabilities of a cache-assisted transmission in the above two cases are different.
- **Density of cache-served requests**. D2D communication usually involves small transmission distances. In networks with high user density, in the case of D2D caching, more simultaneous links are allowed to coexist in the same region sharing the same spectrum resources as compared to the case of SC caching. Moreover, a special case in D2D caching is when a user finds its requested file stored in its own device. In that case, the user request is satisfied without any delay and cost.
- **Power consumption**. Additional to the power consumption required by cache-assisted D2D or small cell transmission, the power cost in the small cell backhaul is also a significant part. When a user request can not be served locally, either through D2D communication or from the SBS caches, the requested content will be retrieved from the core-network through the small cell backhaul. In general the backhaul power consumption is much higher compared to the transmission power cost.

In this chapter, we take into account the aforementioned differences and compare the performances of caching in user devices (i.e., D2D caching) and caching in the SBSs (i.e., SC caching), using models and tools from stochastic geometry. We provide analytical expressions for key performance metrics, including the cache hit probability, the density of cache-served requests and the average power consumption. The numerical results will be presented and compared, which give us insight on the advantages and disadvantages of caching at the mobile devices and at the radio access network edge.

In this section, we address the following question: where should popular content be cached in a wireless network? For that, we model a wireless cellular network using stochastic geometry and analyze the performance of two network architectures, namely caching at the mobile device allowing device-to-device (D2D) connectivity and local caching at the radio access network edge (small cells). We provide analytical and numerical results to compare their performance in terms of the cache hit probability, the density of cache-served requests and average power consumption. Our results reveal that the performance of cache-enabled networks with either D2D caching or small cell caching heavily depends on the user density and the content popularity distribution.

A. Network Model

We consider a small cell network (SCN) where the SBSs are distributed in the two-dimensional Euclidean plane $\mathbb{R}^2$ according to a homogeneous Poisson Point Process (PPP) Φ_{s} with intensity λ_{s}. Mobile users are distributed according to another independent homogeneous PPP Φ_{u} with intensity λ_{u} [35]. Caching capabilities can be enabled either on user devices for potential D2D communication, referred to as D2D caching, or by installing storage units at the SBSs, coined as SC caching.

Each mobile user makes a random request with probability $\rho \in [0,1]$. As a result, the active users to be served form a homogeneous PPP $\Phi_{\mathrm{u}}^{\mathrm{r}}$ with intensity $\rho\lambda_{\mathrm{u}}$ (independent thinning). The inactive users form another homogeneous PPP $\Phi_{\mathrm{u}}^{\mathrm{t}}$ with intensity $(1-\rho)\lambda_{\mathrm{u}}$. They can serve as potential D2D transmitters in the case with D2D caching mode, or remain silent if D2D communication is not enabled.

Depending on whether cache capability is enabled at the devices or at the edge/SBSs, when an active user requests for a file, the following cases may happen:

- with only D2D caching, if the requested file is not cached in its own device, the user searches for the file in the devices in its proximity within a certain distance. If there is at least one potential D2D transmitter that has the requested file cached, the file is transmitted from the nearest one. Otherwise, the user connects to the nearest SBS in order to download the file from the core network through the backhaul.
- with only SC caching, the active user always connects to the nearest SBS. If its associated SBS has the file cached inside, the SBS transmits the file directly to the user. Otherwise the file is downloaded from the core network through the backhaul and then transmitted to the user.

We assume spectrum sharing among concurrent transmissions in such network, i.e. both D2D and small cell communication links receive interference from coexisting transmitters.

1) Request Distribution and Caching Policies: We consider a finite content library $\mathcal{F} = \{f_1, \cdots, f_N\}$ for the user requests, where f_i is the i-th most popular file and N is the library size. All files are assumed to have equal size, which is normalized to one. We use the standard Zipf law for the popularity distribution, meaning that the request probability of the i-th most popular file is

$$p_i = \frac{\Omega}{i^{\gamma}}, \tag{21}$$

where $\Omega = \left(\sum_{j=1}^{N} j^{-\gamma} \right)^{-1}$ is the normalization factor and γ is the shape parameter of Zipf law, which defines the correlation level of user requests. High values of γ means that most of the requests are generated from a few most popular files. For a user making a random request, p_i can be seen as the probability that the requested file is f_i.

Given knowledge of the content popularity distribution, depending on whether D2D caching or SC caching is adopted, we apply the following caching policies:

- with only D2D caching, since a random active user will most likely have multiple potential D2D transmitters, each device will independently cache files subject to its capacity-limited storage according to a common probability distribution, in order to increase the content diversity within the search distance [36]. The optimal caching probabilities are determined by minimizing the average *cache miss* probability;
- with only SC caching, since an active user always connects to the nearest SBS, there are no overlapping coverage areas of different SBSs. Thus, we apply the conventional "cache the most popular content" (MPC) policy, meaning that all SBSs cache the same most popular files within their cache capacity.

The details of the caching policies for both cases are presented as follows.

D2D Caching with Probabilistic Cache Placement. Denote by M_{d} the cache storage size of user devices, where with a random D2D caching policy, each user stores f_i with probability q_i. Let R_{d} be the maximum search/discovery distance of a user device for establishing D2D communication. The probability that no potential D2D transmitters are found when f_i is being requested is equivalent to the probability of having no points from $\Phi_{\mathrm{u}}^{\mathrm{t}}$ with independent thinning probability q_i in the searching area (void probability). The cache miss probability of f_i within the discovery distance R_{d} is given by

$$p_{\mathrm{m}}^{i}(R_{\mathrm{d}}) = e^{-\pi(1-\rho)\lambda_{\mathrm{u}} q_i R_{\mathrm{d}}^2}. \tag{22}$$

The optimal caching probability vector $\mathbf{q} = [q_1, \cdots, q_N]$ for minimizing the average cache miss probability over $\mathcal{F}$ can be found by solving the following optimization problem:

$$\min_{\mathbf{q}} \sum_{i=1}^{N} p_i e^{-\pi(1-\rho)\lambda_u q_i R_{\mathrm{d}}^2} \tag{23}$$

subject to:

$$\sum_{i=1}^{N} q_i - M_{\mathrm{d}} \leq 0 \tag{24}$$

$$q_i \in [0, 1]. \tag{25}$$

We use the optimal dual-solution searching (ODSA) algorithm proposed in [38] in order to find the optimal $\mathbf{q}$. In the remainder of the paper, it is assumed that all the results related to D2D caching are obtained using the optimal caching probabilities.

SC Caching with MPC Policy Denote by M_{s} the cache capacity of a SBS. According to the MPC caching policy, only files f_i with popularity order $i \in [1, M_{\mathrm{s}}]$ would be cached in each SBS.

B. Performance Analysis

The potential gain of wireless content caching is mainly captured by the cache hit probability, which gives opportunity to handle user requests and to deliver content without having to retrieve it from the core network. Furthermore, there are potential spatial reuse gains by establishing proximity-based cache-assisted communication links sharing the same spectrum. In this section, we provide analytical results on several key performance metrics for cache-enabled cellular networks with either D2D caching or SC caching.

1) Cache Hit Probability: The cache hit probability is the probability for a random active user to find its requested file in local caches.

2) D2D Caching: With D2D caching, a cache hit request may happen in two cases:

- a user requesting for a file may find it stored in its own cache, we call this "self-request";
- when the requested file is not cached in its own device, the user finds it in the cache space of its nearby potential D2D transmitters within a distance R_{d}.

Denoting by $p_{\mathrm{self}}^{\mathrm{d}}$ the self-request probability of a random user, we have

$$p_{\mathrm{self}}^{\mathrm{d}} = \sum_{i=1}^{N} p_i q_i. \tag{26}$$

The probability of an active user being served by a nearby potential D2D transmitter is given by

$$p_{\mathrm{r}}^{\mathrm{d}} = \sum_{i=1}^{N} p_i \left(1 - q_i\right) \left(1 - e^{-\pi(1-\rho)\lambda_{\mathrm{u}} q_i R_{\mathrm{d}}^2}\right), \tag{27}$$

where $p_{\mathrm{hit},i}^{\mathrm{d}} = 1 - e^{-\pi(1-\rho)\lambda_{\mathrm{u}} q_i R_d^2}$ is the probability to have at least one potential D2D transmitters within distance R_d with file f_i cached.

Therefore, the cache hit probability is the sum of (26) and (27), given by

$$\begin{aligned} p_{\mathrm{hit}}^{\mathrm{d}} &= p_{\mathrm{self}}^{\mathrm{d}} + p_{\mathrm{r}}^{\mathrm{d}} \\ &= 1 - \sum_{i=1}^{N} p_i \left(1 - q_i\right) e^{-\pi(1-\rho)\lambda_{\mathrm{u}} q_i R_{\mathrm{d}}^2}. \end{aligned} \tag{28}$$

SC Caching The cache hit probability in the case with SC caching is simply the probability that a user finds its requested file stored in the cache of its associated SBS. According to the MPC caching policy, we have that

$$p_{\mathrm{hit}}^{\mathrm{s}} = \sum_{i=1}^{M_{\mathrm{s}}} p_i. \tag{29}$$

3) Success Probability of Cache-assisted Transmission: When an active user finds its requested file cached in either the devices in proximity or the nearest SBS, it is not guaranteed that the cache-assisted transmission of the file will be successful. We calculate here the success probability of a typical cache-assisted transmission conditioning on having a receiver at the origin.

For a given realization of the network, we assume having K established cache-assisted communication links with $\mathbb{E}[K] \leq \rho\lambda_{\mathrm{u}}$. For a random link $i \in [1, K]$, the received signal-to-interference-plus-noise ratio (SINR) is given by

$$\mathrm{SINR}_i = \frac{P_i |h_{i,i}|^2 d_{i,i}^{-\alpha}}{\sigma^2 + \sum_{j \in T \setminus \{i\}} P_j |h_{j,i}|^2 d_{j,i}^{-\alpha}},$$

where $P_i = \{P_{\mathrm{d}}, P_{\mathrm{s}}\}$ denotes the transmit power of either a D2D transmitter or a SBS, depending on whether caching capabilities are enabled on the mobile devices or at the SBSs; $h_{j,i}$ denotes the small-scale channel fading from the transmitter j to the receiver i, which follows $\mathcal{CN}(0, 1)$ (Rayleigh fading); $d_{j,i}$ denotes the distance between the transmitter j and the receiver i; σ^2 denotes the background noise power. Note that interference comes not only from cache-assisted transmissions, but also from the SBS-user links when the requested file of the user is not locally

cached. Thus, in the D2D caching case, the set of active transmitters belong to $T \subseteq \{\Phi_{\text{s}} \cup \Phi_{\text{u}}^{\text{t}}\}$. In the other case with SC caching, $T \subseteq \Phi_{\text{s}}$, because active users can only be served by the SBSs. We assume interference-limited network, in which the background thermal noise is negligible. In the remainder of this analysis the success probability is given as a function of the interference-to-interference ratio (SIR).

D2D Caching. For a random active user requesting for a file, as given in (27), with probability p_{r}^{d} the requested file is not cached in its own device, but in its nearby devices. Therefore, the density of cache-assisted communication links is $\rho\lambda_{\text{u}}p_{\text{r}}^{\text{d}}$. Note that multiple users might find the same nearest D2D transmitter. In that case only one user can connect to this device, others have to search for another D2D transmitter. Denote by $\Phi_{\text{t}}^{\text{d}}$ the set of active D2D transmitters; although the resulting set is not distributed according to a homogeneous PPP, the density of $\Phi_{\text{t}}^{\text{d}}$ is given by

$$\lambda_{\text{t}}^{\text{d}} = \rho\lambda_{\text{u}}p_{\text{r}}^{\text{d}}. \tag{30}$$

The set of users that cannot find their requested files in local caches will be served by the SBSs. The density of users to be served by the SBSs is $\lambda_{\text{r}}^{\text{s}} = \rho\lambda_{\text{u}}(1 - p_{\text{hit}}^{\text{d}})$. According to the nearest SBS association, a Poisson-Voronoi tessellation is generated. From [39], the void probability of a typical Voronoi cell can be approximated as

$$p_{\text{void}} \simeq \left(1 + \frac{\lambda_{\text{r}}^{\text{s}}}{3.5\lambda_{\text{s}}}\right)^{-3.5}. \tag{31}$$

The density of active SBSs is thus given by

$$\begin{aligned} \lambda_{\text{t}}^{\text{s}} &= \lambda_{\text{s}}\left(1 - p_{\text{void}}\right) \\ &\simeq \lambda_{\text{s}}\left(1 - \left(1 + \tfrac{\rho\lambda_{\text{u}}(1-p_{\text{hit}}^{\text{d}})}{3.5\lambda_{\text{s}}}\right)^{-3.5}\right). \end{aligned} \tag{32}$$

Conditioning on having a typical D2D receiver at the origin with its associated transmitter at distance d_x, and assuming a homogeneous PPP for both active D2D transmitters $\Phi_{\text{t}}^{\text{d}}$ and active SBSs $\Phi_{\text{t}}^{\text{s}}$, the success probability is given as [40]

$$\begin{aligned} p_{\text{suc}}^{\text{d}} &= \mathbb{P}\left[\frac{P_{\text{d}}|h_{i,i}|^2 d_x^{-\alpha}}{\sum\limits_{j\in\Phi_{\text{t}}^{\text{d}}\setminus\{i\}} P_{\text{d}}|h_{j,i}|^2 d_{j,i}^{-\alpha} + \sum\limits_{k\in\Phi_{\text{t}}^{\text{s}}} P_{\text{s}}|h_{k,i}|^2 d_{k,i}^{-\alpha}} > \theta\right] \\ &= \mathbb{E}\left[\mathcal{L}_{I_{\text{d}}}\left(\theta d_x^{\alpha}\right)\cdot\mathcal{L}_{I_{\text{s}}}\left(\theta\tfrac{P_{\text{s}}}{P_{\text{d}}}d_x^{\alpha}\right)\right] \\ &= \mathbb{E}\left[\exp\left(-\tfrac{\pi d_x^2\theta^{\frac{2}{\alpha}}}{\text{sinc}(2/\alpha)}\left(\lambda_{\text{t}}^{\text{d}} + \lambda_{\text{t}}^{\text{s}}\left(P_{\text{s}}/P_{\text{d}}\right)^{\frac{2}{\alpha}}\right)\right)\right], \end{aligned} \tag{33}$$

where $\mathcal{L}_{I_{\text{x}}}(s) = \mathbb{E}\left[\exp\left(-sI_{\text{x}}\right)\right]$ is the Laplace transform of interference I_{x} and θ is the SIR threshold for successful D2D transmission. The expectation is over the distribution of d_x and over the content library $\mathcal{F}$.

When a cache hit occurs, the distribution of the D2D link distance d_x depends on the popularity order of the requested file. If f_i is requested by the typical user, conditioning on having at least one potential D2D transmitter within distance R_{d}, the pdf of the D2D link distance d_x is given by

$$f_{d_x}^{i}(r) = \begin{cases} \frac{2\pi(1-\rho)\lambda_{\text{u}}q_i r}{1-e^{-\pi(1-\rho)\lambda_{\text{u}}q_i R_{\text{d}}^2}}e^{-\pi(1-\rho)\lambda_{\text{u}}q_i r^2} & 0 \le r \le R_{\text{d}} \\ 0 & r > R_{\text{d}}. \end{cases} \tag{34}$$

Then we have the approximate success probability when a cache hit of file f_i happens, given by

$$\begin{aligned} p_{\text{suc,i}}^{\text{d}} = \int_0^{R_{\text{d}}} &\exp\left[-\frac{\pi r^2\theta^{\frac{2}{\alpha}}}{\text{sinc}(2/\alpha)}\left(\rho\lambda_{\text{u}}p_{\text{r}}^{\text{d}} + \lambda_{\text{t}}^{\text{s}}\left(P_{\text{s}}/P_{\text{d}}\right)^{\frac{2}{\alpha}}\right)\right] \\ &\cdot\frac{2\pi(1-\rho)\lambda_{\text{u}}q_i r}{1-e^{-\pi(1-\rho)\lambda_{\text{u}}q_i R_{\text{d}}^2}}e^{-\pi(1-\rho)\lambda_{\text{u}}q_i r^2}\text{d}r, \end{aligned} \tag{35}$$

where p_{r}^{d} and $\lambda_{\text{t}}^{\text{s}}$ are given in (27) and (32), respectively.

SC Caching. In the case with SC caching, users are always connected to the nearest SBSs in both cache hit and cache miss events. Denote by $\tilde{\Phi}_{\text{t}}^{\text{s}}$ the set of active SBSs, similarly, using the void probability of Voronoi cell in

(31), the density of active SBSs is given by

$$\tilde{\lambda}_{\mathrm{t}}^{\mathrm{s}} \simeq \lambda_{\mathrm{s}} \left(1 - \left(1 + \frac{\rho\lambda_{\mathrm{u}}}{3.5\lambda_{\mathrm{s}}}\right)^{-3.5}\right). \tag{36}$$

Conditioning on having the typical receiver at the origin with its associated SBS at distance d_s and using nearest SBS association, we have the pdf of d_s given as

$$f_{d_s}(r) = 2\pi\lambda_{\mathrm{s}} r \cdot e^{-\pi\lambda_{\mathrm{s}} r}. \tag{37}$$

For a given SIR threshold θ, the success probability of the cache-assisted small cell transmission is given by

$$\begin{aligned} p_{\mathrm{suc}}^{\mathrm{s}} &= \mathbb{P}\left[\frac{P_{\mathrm{s}}|h_{i,i}|^2 d_s^{-\alpha}}{\sum\limits_{k\in\tilde{\Phi}_{\mathrm{t}}^{\mathrm{s}}\setminus\{i\}} P_{\mathrm{s}}|h_{k,i}|^2 d_{k,i}^{-\alpha}} > \theta\right] \\ &= \int_0^\infty 2\pi\lambda_{\mathrm{s}} r \cdot e^{-\pi\lambda_{\mathrm{s}} r^2} \exp\left(-\frac{\pi\tilde{\lambda}_{\mathrm{t}}^{\mathrm{s}} r^2 \theta^{\frac{2}{\alpha}}}{\mathrm{sinc}(2/\alpha)}\right) \mathrm{d}r, \end{aligned} \tag{38}$$

where $\tilde{\lambda}_{\mathrm{t}}^{\mathrm{s}}$ is given in (36).

4) Density of Cache-served Requests: A user request is said to be "served" if the requested file is found in local caches and if the transmission of the file is successful. Based on the above results, we calculate the density of cache-served requests, which is the average number of requests that can be successfully and simultaneously handled by the local cache per unit area.

D2D Caching In the D2D caching case, a random user request can be served either by self-request or through proximal D2D communication. Denote by $\mu_{\mathrm{suc}}^{\mathrm{d}}$ the density of cache-served requests, we have

$$\begin{aligned} \mu_{\mathrm{suc}}^{\mathrm{d}} &= \rho\lambda_u \left(p_{\mathrm{self}}^{\mathrm{d}} + \sum_{i=1}^{N} p_i(1-q_i) p_{\mathrm{hit},i}^{\mathrm{d}} p_{\mathrm{suc,i}}^{\mathrm{d}}\right) \\ &= \rho\lambda_u \sum_{i=1}^{N} p_i \left[q_i + (1-q_i)\left(1 - e^{-\pi(1-\rho)\lambda_{\mathrm{u}} q_i R_d^2}\right) p_{\mathrm{suc,i}}^{\mathrm{d}}\right], \end{aligned} \tag{39}$$

where $p_{\mathrm{suc,i}}^{\mathrm{d}}$ is given in (35).

SC Caching In the SC caching case, the maximum number of cache-assisted transmissions in a given time slot is limited by the density of the SBSs. The probability for a SBS to have at least one active user in its cell requesting for files that are stored in its cache is given by

$$p_{\mathrm{t}}^{\mathrm{s}} = 1 - \left(1 + \frac{\rho\lambda_{\mathrm{u}} p_{\mathrm{hit}}^{\mathrm{s}}}{3.5\lambda_{\mathrm{s}}}\right)^{-3.5}. \tag{40}$$

The density of cache-assisted transmission is $\lambda_{\mathrm{s}} p_{\mathrm{t}}^{\mathrm{s}}$. Then, the density of cache-served requests, which is the density of successful cache-served small cell transmission, is given by

$$\mu_{\mathrm{suc}}^{\mathrm{s}} = \lambda_{\mathrm{s}} p_{\mathrm{t}}^{\mathrm{s}} p_{\mathrm{suc}}^{\mathrm{s}} = \lambda_{\mathrm{s}} \left(1 - \left(1 + \frac{\rho\lambda_{\mathrm{u}} p_{\mathrm{hit}}^{\mathrm{s}}}{3.5\lambda_{\mathrm{s}}}\right)^{-3.5}\right) p_{\mathrm{suc}}^{\mathrm{s}}, \tag{41}$$

where $p_{\mathrm{suc}}^{\mathrm{s}}$ is given in (38).

5) Power Consumption: For a random user request, in a cache hit event, the consumed power for content delivery contains only the transmit power of either the D2D transmitter or the associated SBS. In a cache miss event, the requested file is first fetched from the core network via the backhaul and then transmitted from the nearest SBS to the user. Thus, additional energy is consumed at the backhaul. Denote P_{b} the backhaul power consumption required to handle a user request at a single SBS; we study below the power consumption per user request with either D2D caching or SC caching.

For the case with D2D caching, recall that when self-request occurs, no energy is consumed to serve the request. A random user request has probability $p_{\mathrm{r}}^{\mathrm{d}}$ to be served by a nearby D2D transmitter, and probability $1 - p_{\mathrm{hit}}^{\mathrm{d}}$ to be served by the nearest SBS. We have the consumed power per user request given as

$$P_{\mathrm{avg}}^{\mathrm{d}} = p_{\mathrm{r}}^{\mathrm{d}} P_{\mathrm{d}} + (1 - p_{\mathrm{hit}}^{\mathrm{d}})(P_{\mathrm{s}} + P_{\mathrm{b}}). \tag{42}$$

For the case with SC caching, when a random user request occurs, the transmission power of its nearest SBS is always consumed for both cache hit and cache miss events. The backhaul power is additionally consumed with probability $1 - p_{\text{hit}}^{\text{s}}$. Thus we have

$$P_{\text{avg}}^{\text{s}} = P_{\text{s}} + (1 - p_{\text{hit}}^{\text{s}})P_{\text{b}}. \tag{43}$$

C. Numerical Results

For numerical evaluation, we set the SBS density at $\lambda_{\text{s}} = 10^{-5}$ and the user density at $\lambda_{\text{s}} = K \times 10^{-4}$, where K is a proportion factor. We choose $\rho = 0.2$, meaning that 20% of the users will request for a file in the content library $\mathcal{F}$. The cache storage size at a SBS and at a user device are $M_{\text{s}} = 10^4$ and $M_{\text{d}} = 10$, respectively. The content library has size $N = 10^5$, with popularity distribution given by Zipf law with shape parameter $\gamma = 0.7$ for the case with low popularity skewness, and $\gamma = 1.2$ for the other case. The searching distance of a user device to establish D2D link is $R_{\text{d}} = 75$ m. The transmit power of a SBS and a user device are $P_{\text{s}} = 100$ mW and $P_{\text{d}} = 2$ mW, respectively. Backhaul power consumption at the SBS to handle a user request is $P_{\text{b}} = 10P_{\text{s}} = 1$ W. The target SIR of successful transmissions is chosen as $\theta = 0$ dB. We present numerical results for $K \in [1, 9]$, in order to compare the performance between caching at the user devices and caching at the SBSs for different user density regimes.

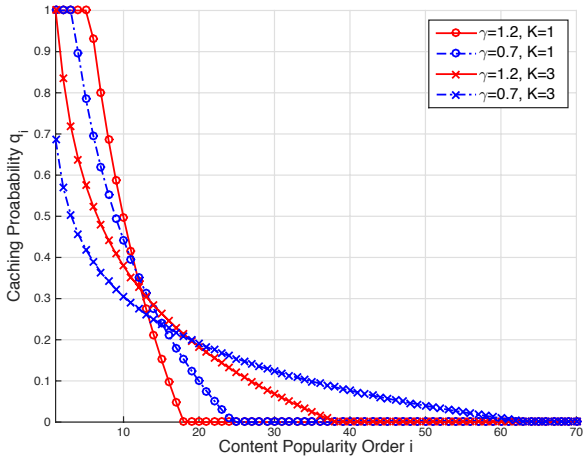

Fig. 6. Optimal caching probability vector $\mathbf{q} = [q_1, \cdots, q_N]$.

In Fig. 6, we plot the optimal caching probability q_i of file f_i, $i \in [1, N]$, obtained by solving the optimization problem in (23). Comparing the results with $\gamma = 1.2$ and with $\gamma = 0.7$, we see that for lower γ, users tend to cache more different files, because the user requests are more diverse due to the low popularity concentration level. We observe the same trend when user density is higher, e.g., $K = 3$. This is reasonable because with higher user density the probability to establish D2D communication is higher. Thus, more different files should be cached in user devices in order to serve more requests by cache-assisted D2D transmission.

1) Cache Hit Probability: Fig. 7 shows the cache hit probability obtained with (28) and (29) for the case with D2D caching and with SC caching, respectively. As expected, the cache hit probability for both cases are higher with lower γ. Furthermore, we see that caching at the SBSs results in much higher cache hit probability than caching at the mobile devices, as a result of the larger cache capacity.

2) Density of Cache-served Requests: The density of cache-served requests measures how many requests can be successfully handled simultaneously using the local caches. From Fig. 8, we see that D2D caching outperforms SC caching for higher γ, especially in the high user density regime. In the case with lower γ, the performance of

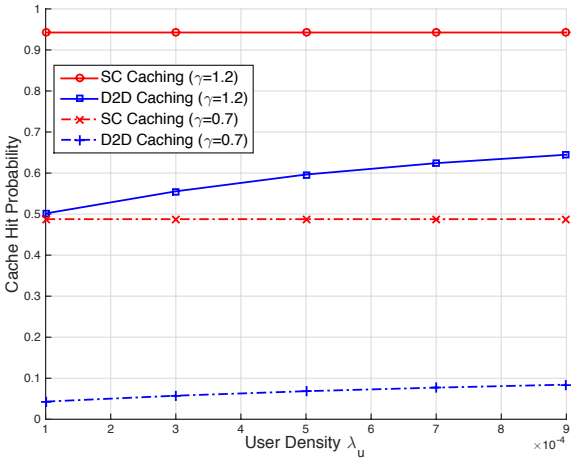

Fig. 7. Cache hit probability vs. user density λ_s.

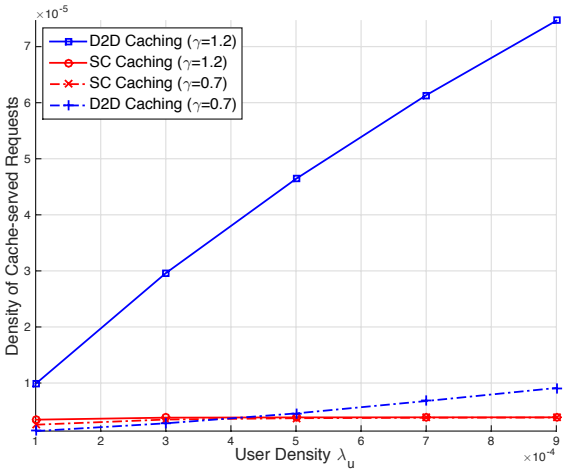

Fig. 8. Density of cache-served requests vs. user density λ_s.

SC caching is slightly better in the sparse user regime, whereas the performance of D2D caching outperforms SC caching when the user density increases. The advantage of D2D caching is mainly due to two reasons:

- when the user density is high, the number of potential D2D transmitters are in general larger than the number of SBSs, which allows to have more simultaneous cache-assisted transmission links;
- self-request of D2D caching gives opportunity to handle a large amount of requests made by the users when the content popularity is highly concentrated.

3) Power Consumption: Fig. 9 plots the average power consumption per user request, showing that SC caching is expectedly more energy efficient than the case with D2D caching, which is mainly because of the high cache hit probability. Compared to the consumed power for fetching the content through the backhaul, the transmit power of both user devices and SBSs are relatively low. Hence, the higher the probability to serve a user request locally, the less power is needed. We also observe that the power consumption per user request with D2D caching decreases with the user density. This is because the number of potential D2D transmitters within the discovery distance of a user increases when the user density increases, thus giving higher probability to serve the user request by cache-assisted D2D transmission.

Combining these results, we have the following takeaway messages:

- in networks with high user density, D2D caching gives the opportunity to serve more user requests simulta-

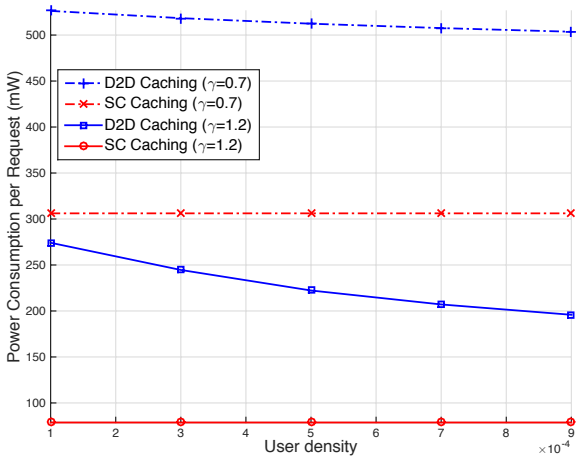

Fig. 9. Power consumption per user request vs. user density λ_s.

neously through short-distance cache-assisted D2D communication;

- edge (SBSs) caching results in much higher cache hit probability because storage units at the SBSs have much larger capacity than at the mobile devices. As a result, SC caching is also more energy efficient because less power is consumed on average at the backhaul in order to download a file from the core network.

D. Conclusions

We compared the performance of D2D caching and small cell caching in a spatial wireless network. We provided analytical expressions, corroborated by numerical results, for the cache hit probability, the density of cache-served requests and the power consumption. The main takeaway message is that in wireless networks with high user density, it is beneficial in terms of the spatial reuse of communication resources to perform caching at the mobile devices and to benefit from proximal D2D communication. On the other hand, deploying large-capacity storage units for caching at the edge (i.e. in small cells) is more advantageous in terms of energy efficiency, as a result of the high cache hit probability in such case.

V. Research Challenges for Wireless Caching

In this chapter we gave mathematical evidence of the caching benefits for the operation of future wireless networks. We complete the chapter by briefly summarizing the most challenging problems and the most interesting future directions of research for caching in wireless networks. The interested reader is also referred to [41]–[44].

A. Modelling the requests in wireless networks

The standard model for performance analysis of web caching is the Independence Reference Model (IRM): content m is requested according to an independent Poisson process with rate λp_m, where p_m refers to the content popularity, modelled by a power law, i.e., $p_m \propto m^{-\alpha}$. This well-established model thrives due to its simplicity; it only has two parameters, namely λ to control the rate of requests, and α to control the skew of the popularity.

The IRM assumes that the content popularity is static, which is of course not true. Breaking news, and the episodes of TV series are examples of ephemeral content with rapidly changing popularity; they appear, they become increasingly popular, and they gradually become unpopular again. Recent works [45], [46] propose novel mathematical tools to model time-varying content popularity. The studies consider large samples of YouTube and VoD applications and discover: content popularity is time-varying and in addition has a consistent effect on analyzing the caching performance. In the inhomogeneous Poisson model proposed in [45] each content is associated to a shot whose duration reflects the content lifespan and whose height denotes its instantaneous popularity. The model is called the Shot Noise Model (SNM), mirroring the Poisson noise from electronics. While the shape of the pulse is not important, the study [46] observes strong correlations between popularity and duration; apparently popular

contents prosper longer. Finally, a class-based model [45] can conveniently capture spatio-temporal correlations while allowing analytical tractability. Mobile users are profound downloaders of ephemeral content, thus it is expected that a similar analysis for the case of wireless content will show stronger effects.

In fact, the concept of time-varying popularity is not entirely new to caching engineers. Currently, web caching systems operate with dynamic eviction policies like Least-Recently-Used (LRU) and Least-Frequently-Used (LFU) which combat time-varying content popularity in a heuristic manner. The performance of LRU under the SNM model is analyzed in [47]. A recent study [48] analyzes the SNM model and gives the optimal policy for joint caching and popularity estimation. The result shows that (LFU-like) frequency estimation becomes optimal for contents with more than a few requests. Continuously estimating the popularity is more difficult in wireless systems where the requests of the population are split over multiple caches and those that reach any given cache are less. This development motivates novel caching techniques that employ learning methodologies to accurately track the evolution of content popularity over time.

B. Coded caching for broadcast medium exploitation

The fundamental work of Maddah-Ali and Niesen [49] proposed the use of caching to improve the efficiency of wireless transmissions in a cell. While traditional techniques cache entire contents at mobile terminals to eliminate retransmissions of popular content, the coded caching technique stores coded combinations of content chunks. Then, the base station can broadcast a coded sequence of messages so that all receivers in the group can combine it with the cached chunks to decode the contents. This is shown to yield resource blocks equal to $N(1-K/M)/(1+NK/M)$, where N is the number of users, K the cache size, and M the catalog size. Hence, if the cacheable fraction of the catalog K/M is kept fixed, then the required number of resource blocks does not increase with the number of users N, this can be verified by taking the limit $N \to \infty$ whereby the above quantity converges to a constant. In other words, in a scenario of a wireless base station with N users watching individual content, the result shows that we may increase N indefinately while maintaining a fixed service quality.

This revolutionary idea has sprung a wealth of research efforts, such as device-to-device (D2D) networks [50], [51], non-uniform content popularities [52], online caching policies [53], multi-servers [54], multi-library [55], and combination with CSI [56]. From the implementation point of view, promising research directions include extensions to capture system aspects such as (i) popularity skew, (ii) asynchronous requests, (iii) finite code lengths and (iv) cache sizes that scale slower than M. Assuming that these practical challenges are resolved, caching for wireless systems will become intertwined with physical layer techniques employed at the base station and the handheld.

C. Cooperative caching models

In the context of heterogeneous wireless networks, a user can connect in a variety of ways to the network in order to obtain the requested content. For example, contemporary smart phones receive the signal of more than 10 base stations simultaneously. In future densified cellular networks, the mobile will be connected to several femto-, pico-, or nano- cells. The phenomenon of wireless multi-access opens a new horizon in caching exploitation [32]. Since a user can retrieve the requested content from many network endpoints, neighbouring caches should collaborate and avoid storing the same objects multiple times.

Most content placement optimizations of wireless caching boil down to a set cover problem in a bipartite graph connecting the users to the reachable caches. Therefore, finding what contents to store at each cache is a difficult problem even if the popularities are assumed known [32]. It is possible to relax the problem to convex optimization by the use of distributed storage codes, where each cache stores coded combinations of contents [32], or by time sharing placement of different contents. These ideas lead to several interesting algorithms in the literature of multi-access caching [57]–[60].

REFERENCES

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update 2014–2019," *White Paper*, 2015. [Online]. Available: http://goo.gl/tZ6QMk

[2] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, pp. 388–404, Mar. 2000.

[3] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.

[4] G. S. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *CoRR*, vol. abs/1602.00173, 2016.

[5] M. Franceschetti and R. Meester, *Random Networks for Communication*. New York, NY, USA: Cambridge University Press, Series: Cambridge Series in Statistical and Probabilistic Mathematics (No. 24), 2007.

[6] J. Roberts and N. Sbihi, "Exploring the memory-bandwidth tradeoff in an information-centric network," *CoRR*, vol. abs/1309.5220, 2013. [Online]. Available: http://arxiv.org/abs/1309.5220

[7] S. Gitzenis, G. S. Paschos, and L. Tassiulas, "Asymptotic laws for content replication and delivery in wireless networks," in *Proceedings of the 2012 IEEE INFOCOM conference on Computer Communications*, Orlando, FL, USA, Mar. 2012, pp. 126–134.

[8] S. Gitzenis, G. Paschos, and L. Tassiulas, "Asymptotic laws for joint content replication and delivery in wireless networks," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2760–2776, 2013.

[9] G. S. Paschos, S. Gitzenis, and L. Tassiulas, "The effect of caching in sustainability of large wireless networks," in *Proceedings of the 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2012),*, may 2012, pp. 355 –360.

[10] S. Gitzenis, G. S. Paschos, and L. Tassiulas, "Enhancing wireless networks with caching: Asymptotic laws, sustainability & tradeoffs enhancing wireless networks with caching: Asymptotic laws, sustainability & trade-offs," in *accepted for publication in Computer Networks*, 2014.

[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4. ACM, 1999, pp. 251–262.

[12] M. G. G. Alfano and E. Leonardi, "Content-centric wireless networks with limited buffers: when mobility hurts," in *Proceedings of 2013 IEEE INFOCOM conference on Computer Communications*, Torino, Italy, Apr. 2013.

[13] U. Niesen, D. Shah, and G. Wornell, "Caching in wireless networks," in *Proceedings IEEE International Symposium on Information Theory*, Seoul, Korea, Jun. 2009, pp. 2111–2115.

[14] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network," in *Proceedings on the 2012 IEEE INFOCOM conference on Computer Communications Workshops*. IEEE, 2012, pp. 310–315.

[15] T. Bektas, J.-F. Cordeau, E. Erkut, and G. Laporte, "Exact algorithms for the joint object placement and request routing problem in content distribution networks," *Computers & Operations Research*, vol. 35, no. 12, pp. 3860 – 3884, 2008.

[16] J. Munoz-Gea, S. Traverso, and E. Leonardi, "Modeling and evaluation of multisource streaming strategies in P2P VoD systems," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 4, pp. 1202–1210, 2012.

[17] V. Lenders, G. Karlsson, and M. May, "Wireless ad hoc podcasting," in *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2007*. IEEE, 2007, pp. 273–283.

[18] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT) 2012*. IEEE, 2012, pp. 2781–2785.

[19] X. Jiang, P. Gao, Y. Zhao, and Y. Shi, "Caching scheme research based on unstructured peer-to-peer network," *Physics Procedia*, vol. 25, pp. 1076–1083, 2012.

[20] C. Liaskos, S. Petridou, G. Papadimitriou, P. Nicopolitidis, M. Obaidat, and A. Pomportsis, "A novel clustering-driven approach to wireless data broadcasting," in *Proceedings of the IEEE/CVR 15th Annual Symposium*, 2011.

[21] C. R. Cunha, A. Bestavros, and M. E. Crovella, "Characteristics of WWW client-based traces," in *View on NCSTRL*, Boston University, MA, USA, Jul. 1995.

[22] L. Breslau, P. Cue, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of the 1999 IEEE INFOCOM conference on Computer Communications*, New York, NY, USA, Mar. 1999, pp. 126–134.

[23] L. A. Adamic and B. A. Huberman, "Zipf's law and the Internet," *Glottometrics*, vol. 3, pp. 143–150, 2002.

[24] V. N. Padmanabhan and L. Qiu, "The content and access dynamics of a busy web site: Findings and implications," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 111–123, 2000.

[25] T. Yamakami, "A Zipf-like distribution of popularity and hits in the mobile web pages with short life time," in *Proceedings of Parallel and Distributed Computing, Applications and Technologies, PDCAT '06*, Taipei, ROC, Dec. 2006, pp. 240–243.

[26] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 1–14.

[27] G. Dán and N. Carlsson, "Power-law revisited: A large scale measurement study of P2P content popularity," in *Proceedings of the International Workshop on Peer-To-Peer Systems (IPTPS), San Jose, CA, USA*, 2010.

[28] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng, "Understanding user behavior in large-scale video-on-demand systems," in *ACM SIGOPS Operating Systems Review*, vol. 40, no. 4. ACM, 2006, pp. 333–344.

[29] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *CoRR*, vol. abs/1209.5807, 2012.

[30] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *CoRR*, vol. abs/1405.5336, 2014.

[31] E. Bastug, M. Bennis, M. Kountouris, and M. Debbah, "Cache-enabled small cell networks: modeling and tradeoffs," *EURASIP Journal on Wireless Commun. and Networking*, vol. 2015, no. 1, pp. 1–11, Feb. 2015.

[32] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc., IEEE INFOCOM*, Mar. 2012, pp. 1107–1115.

[33] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-aho, "Content-aware user clustering and caching in wireless small cell networks," in *IEEE Intl. Symposium on Wireless Commun. Systems (ISWCS)*, Aug. 2014, pp. 945–949.

[34] M. Ji, G. Caire, and A. Molisch, "Wireless device-to-device caching networks: basic principles and system performance," *IEEE Journal on Sel. Areas in Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[35] A. Baddeley, *Spatial Point Processes and their Applications*. Springer Verlag, 2007, vol. 1892, pp. 1–75.

[36] M. Ji, G. Caire, and A. Molisch, "Optimal throughput-outage trade-off in wireless one-hop caching networks," in *Proc., IEEE Intl. Symp. on Inform. Theory (ISIT)*, Jul. 2013, pp. 1461–1465.

[37] N. Golrezaei, A. Dimakis, and A. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Trans. on Inform. Theory*, vol. 60, no. 7, pp. 4286–4298, Jul. 2014.

[38] H. Kang, K. Park, K. Cho, and C. Kang, "Mobile caching policies for device-to-device (D2D) content delivery networking," in *Proc., IEEE Conf. on Computer Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 299–304.

[39] S. Lee and K. Huang, "Coverage and economy of cellular networks with many base stations," *IEEE Commun. Letters*, vol. 16, no. 7, pp. 1038–1040, Jul. 2012.

[40] M. Haenggi and R. K. Ganti, "Interference in large wireless networks," *Found. Trends Netw.*, vol. 3, no. 2, pp. 127–248, Feb. 2009.

[41] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.

[42] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug 2014.

[43] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16–22, August 2016.

[44] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 22–28, September 2016.

[45] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, "Temporal locality in today's content caching: Why it matters and how to model it," *ACM SIGCOMM*, 2013.

[46] F. Olmos, B. Kauffmann, A. Simonian, and Y. Carlinet, "Catalog dynamics: Impact of content publishing and perishing on the performance of a LRU cache," in *Teletraffic Congress (ITC), 2014 26th International*, Sept 2014, pp. 1–9.

[47] E. Leonardi and G. L. Torrisi, "Least recently used caches under the shot noise mode," in *IEEE INFOCOM*, 2015, pp. 2281–2289.

[48] M. Leconte, G. S. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, "Placing dynamic content in caches with small population," in *IEEE INFOCOM*, 2016.

[49] M. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[50] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *arXiv preprint arXiv: 1405.5336*, 2014.

[51] S.-W. Jeon, S.-N. Hong, M. Ji, and G. Caire, "On the capacity of multihop device-to-device caching networks," in *IEEE Information Theory Workshop (ITW)*. IEEE, 2015, pp. 1–5.

[52] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *arXiv preprint arXiv: 1502.03124*, 2015.

[53] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *arXiv preprint arXiv: 1311.3646*, 2013.

[54] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *arXiv preprint arXiv: 1503.00265*, 2015.

[55] S. Sahraei and M. Gastpar, "Multi-library coded caching," *arXiv preprint arXiv:1601.06016*, 2016.

[56] J. Zhang and P. Elia, "Fundamental limits of cache-aided wireless BC: Interplay of coded-caching and CSIT feedback," *arXiv preprint arXiv: 1511.03961*, 2015.

[57] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3665–3677, 2014.

[58] K. Naveen, L. Massoulie, E. Baccelli, A. Carneiro Viana, and D. Towsley, "On the interaction between content caching and request assignment in cellular cache networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2015, pp. 37–42.

[59] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *IEEE INFOCOM*, 2015.

[60] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Communications (ICC), 2015 IEEE International Conference on*, June 2015, pp. 3358–3363.