Optimal Cache Allocation for Femto Helpers with Joint Transmission Capabilities

Alina Tuholukova, alina.tuholukova@inria.fr, Giovanni Neglia, giovanni.neglia@inria.fr,
Thrasyvoulos Spyropoulos, thrasyvoulos.spyropoulos@eurecom.fr

***Abstract*—As cellular network operators are struggling to keep up with the rapidly increasing traffic demand, two key directions are deemed necessary for beyond 4G networks: (i) extensive cell densification to improve spatial reuse, and (ii) storage of content as close to the user as possible to cope with the backhaul constraints and increased interference. However, caching has mostly been studied with an exclusive focus either on the backhaul network (e.g. the "femto-caching" line of work) or on the radio access (e.g. through coded caching or cache-aided CoMP). As a result, an understanding of the impact of edge caching on network-wide and end-to-end performance is lacking. In this paper we investigate the problem of optimal caching in a context where nearby small cells ("femto-helpers") can coordinate not just in terms of what to cache but also to perform Joint Transmission (a type of CoMP). We show that interesting tradeoffs arise between caching policies that improve radio access and ones that improve backhaul, and propose an algorithm that provably achieves an 1/2-approximation ratio to the optimal one (which is NP-hard), and performs well in simulated scenarios.**

***Index Terms*—caching, joint transmission, CoMP, heterogeneous cellular networks**

I. Introduction

Traffic demand in cellular networks continues to increase at high pace. The prediction is that by 2020 mobile data traffic will reach 30.6 exabytes per month, 75% of which will be video related [1]. To keep up with this trend, it is widely accepted that the mobile network has to become considerably denser and heterogeneous, with overlapping layers of small cells (e.g. pico, femto). This densification promises to considerably increase the rates offered to users over the air. However, it also poses a significant challenge on the design of the backhaul network, which now has to carry significantly more traffic per m^2 over capacity limited and usually wireless links, threatening to become the new bottleneck.

To this end, researchers have proposed to push popular content closer to the user (e.g. at small cells) during off-peak hours, in order to reduce backhaul traffic at peak hours, and also to reduce the access latency to the content. One of the first works to study the problem of edge caching is [2], which coined the term *femto-caching*. The paper considers a dense network, where a user can communicate with multiple base stations with local caches. When a user's requested content is available at the reachable base stations, the request is satisfied by the base station with the highest transmission rate, without adding load to the backhaul. Hence, unlike the case of isolated caches, where it's optimal to cache the most popular files in each, caching different files at the base stations can increase the amount of *total* cache space accessible to a user leading to better hit rates. However, as different users might see different base stations, with partially overlapping coverage, the problem of whether to cache the same or different files becomes hard, and the authors propose efficient approximation algorithms.

A number of follow-up works have extended the femto-caching framework, e.g. for storage in user devices [3], multi-layer video streaming where video quality can be traded off with hit rate [4], multicast through multiple helper nodes, using LTEs eMBMS framework [5], considering social aspects [6], as well as dynamic cache replacement policies [7], [8] and the tradeoff between edge caching and user request routing Naveen et al. [9]. The common denominators between most of these works can be summarized as follows: (i) The main bottleneck is the backhaul link, (ii) the transmission phase is ignored (assuming requests are asynchronous and non-interfering) or simplified, (iii) global caching gains stem from cells coverage overlaps.

Nevertheless, when considering a wireless setup, content delivery over the radio access link becomes just as important as the placement problem. If multiple nearby base stations (BS) have the same content cached, they can coordinate in order to improve performance on the radio access link. For example, several base stations can perform Joint Transmission (JT) to simultaneously transmit the same file to a single user, e.g. for power and diversity gains, which is particularly useful to edge users. Alternatively, multiple user requests could be satisfied in parallel by forming a MU-MIMO channel, between the BSs and users involved. Such techniques are often referred to as Coordinated Multi-point (CoMP) transmission [10]. With the data cached locally on each BS involved, only channel state information (CSI) information needs to be exchanged over the backhaul to coordinate the transmission, which is a much smaller burden, compared to exchanging whole video files. These ideas have led researchers to argue that caching and transmission algorithms at each involved BS must be jointly designed in order to facilitate such CoMP opportunities, whether this is a distributed BS setup [12] or cloudRAN scenario [13].

Recent work by Maddah-Ali and Niesen [11] revealed quite interesting findings about the fundamental gains achievable by jointly considering caching and coded transmissions in a broadcast channel. Finally, in the very recent work of [14], ideas from coded caching are also used to derive fundamental performance bounds on the impact of caching for a simple K-user interference channel.

A dichotomy appears then in the existing literature: the

femto-caching line of work aims to reduce backhaul traffic and then focuses on hit rates as main performance metric, the cache-aided communication line of work instead maximizes the transmission rates achievable on the radio access channel, ignoring the effect of cache misses. As a result, a clear understanding of the impact on end-to-end (or network-wide) performance is lacking and our paper targets this omission. We address the problem of cache placement that jointly optimizes both radio access and backhaul performance, in a setup where small cells ("femto-nodes") can coordinate both in terms of what they cache and in how they transmit.

As a first step in this direction, we focus on JT technique for the radio access part. Every time the requested file is cached at several base stations in the user's range, the base stations can jointly transmit the file to the user. The transmission rate of JT is higher than that of each separate base station. Hence, storing the same (popular) files is optimal with respect to radio access transmission. On the other hand, storing different files in these base stations might lead to fewer cache misses and thus accesses to the backhaul network, which is important if the latter is the bottleneck. To the best of our knowledge, the only other paper looking at the radio access/backhaul tradeoff is [15], where two different CoMP techniques are studied, namely Maximum Ratio Transmission (MRT) and Zero-Forcing BeamForming (ZFBF) [16]. The authors consider two caching heuristics: a randomized caching policy for MRT and a threshold policy for ZFBF. While they derive the optimal parameter setting of such heuristics, they are in general suboptimal and there is no theoretical performance guarantee in comparison to the optimal content allocation. On the contrary, our allocation algorithm has a provable approximation ratio.

More in details, in this paper we make the following contributions:

- We formulate the problem of optimal cache placement towards optimizing *end-to-end* content download delay;
- We show that the problem is NP-hard, but has desirable submodularity properties that lead to an efficient algorithm with a provable $1/2$-approximation ratio.
- We compare our scheme towards standard femto-caching (where a user fetches a content from the best BS), as well as an "advanced" femto-caching policy where caching is performed as in the baseline, but opportunities for JT transmissions are exploited. Our findings suggests our joint policy can best exploit the tradeoffs existing in this context, and also reveals some interesting tradeoffs in different operating regimes.

The rest of this paper is structured as follows. First, in Section II we formulate the problem of caching that minimizes the downloading time of a file as an integer programming problem. Section III discusses the particular case of the problem, when the signal that all users receive from the base stations is the same. We show, that under some conditions, the target function is submodular over matroid constraints, what means that the problem can be solved efficiently with guaranteed approximation. Then in Section IV we discuss what changes for the general model. In Section V we present the simulation results. We find the cases for which the proposed caching has significant benefits comparing to the femto-caching policy.

II. Problem modeling

Content: There is a content catalog of F files, and file f is characterized by popularity p_f, e.g. expressed as the request rate for file f. For simplicity we assume that all the files are of the same size M (for example, we can think that the files are split in chunks of the same size).

Network nodes: We assume there are H small cells (micro/pico/femto), which we will call helpers, each able to store up to C different files. Again, for simplicity, we assume the same caching size for all helpers. The problem can be easily generalized to the case when the caches of the helpers are of different sizes. There are also U users each requesting a file according to the above popularity. We assume that the requests are asynchronous.

User-Helper Connectivity: We introduce the variables e_{iu} denoting if user u can download from helper i ($e_{iu} = 1$) or not ($e_{iu} = 0$). When the download is possible g_{iu} denotes the corresponding downlink SNR. For convenience, we also define $g_{iu} = 0$ when u cannot download from i.

Storage Variables: Let $\mathbf{X}$ be a $H \times F$ matrix that tracks which files are cached on which helpers, thus $x_{if} = 1$ if the file f is cached on the helper i, and $x_{if} = 0$ otherwise. We will call $\mathbf{X}$ the placement or caching matrix. $\mathbf{X}$ is the main control variable in our system and our goal is to choose $\mathbf{X}$ optimally to minimize the average end-to-end delay per content download.

Let $k(u, f)$ be the number of copies of the file f in the helpers of the user u under placement $\mathbf{X}$:

$$k(u, f) = \sum_{i=1}^{H} x_{if} e_{iu}.$$

All the notations are summarized in Table I.

Performance metric: We focus on downlink traffic and consider the end-to-end delay to fetch a content which consists of two components: (i) the *backhaul delay*, which has to be incurred only if the content is not locally cached; (ii) the *radio access delay*, which depends on the transmission policy assumed. In the following we detail our assumptions about the radio access transmission, and how these affect the above two delay components.

A. Non-cooperative (baseline) transmission

In this simple setup, we assume that base stations do not cooperate during transmission. This is the case for the basic femto-caching setup [2]. Without loss of generality, we will use the Shannon rate for radio transmissions.

Cache miss: If the content is not found in any nearby BS, i.e. $k(u, f) = 0$, then it is fetched over the backhaul to the BS with the best signal for user u, incurring a fixed delay d_b. It is then transmitted from that BS only, incurring *radio access delay* equal to:

$$d_{r,m}(u) = \frac{M}{W \log_2 \left(1 + \max\limits_{i=1,\dots H} g_{iu}\right)},$$

and a total delay equal to $d_m(u) = d_{r,m}(u) + d_b$.

Cache hit: If the requested content is found on at least one helper within the transmission range, i.e. $k(u,f) > 0$, then it is downloaded by the one with the best channel, and the *radio access delay* in this non-cooperative setting is given by:

$$d_{r,h}^{(nc)}(u,f) = \frac{M}{W \log_2 \left(1 + \max\limits_{i=1,\dots H} g_{iu} x_{if}\right)}.$$

The *backhaul delay* in this case is 0.

Optimization problem: Our goal is to find the optimal allocation that minimizes the average delay in the network $\bar{d}^{(nc)}$:

$$\begin{aligned}
&\bar{d}^{(nc)} = \\
&= \sum_{u,f} p_f \left(1_{k(u,f)=0} d_m(u) + 1_{k(u,f)>0} d_{r,h}^{(nc)}(u,f)\right) \\
&= \sum_{u,f} p_f \left(\left(1 - 1_{k(u,f)>0}\right) d_m(u) + 1_{k(u,f)>0} d_{r,h}^{(nc)}(u,f)\right) \\
&= \sum_{u,f} p_f \left(d_m(u) - 1_{k(u,f)>0} \left(d_m(u) - d_{r,h}^{(nc)}(u,f)\right)\right) \\
&= \sum_{u} d_m(u) - \sum_{u,f} p_f 1_{k(u,f)>0} \left(d_m(u) - d_{r,h}^{(nc)}(u,f)\right).
\end{aligned}$$

Observing that the first term $\sum_u d_m(u)$ does not depend on content allocation $\mathbf{X}$ minimizing the delay is equivalent to the following problem:

Problem 1 *(Femto problem): In a non-cooperative setting, minimizing the average delay is equivalent to solve the following maximization problem:*

maximize:

$$F^{(nc)}(\mathbf{X}) = \sum_{u,f} p_f 1_{k(u,f)>0} \left(d_m(u) - d_{r,h}^{(nc)}(u,f)\right)$$

subject to: $\sum_{f=1}^{F} x_{if} \leq C$, for $i = 1, \dots H$

Problem 1 is an integer programming problem and it is equivalent to the original femto-caching problem considered in [2]. The only difference is the interpretation of the delay upon a miss $d_m(u)$. Here, it is the sum of the backhaul delay and a radio delay, while in [2] it is the retrieval time from a slow macro-BS. From the results in [2] it follows then that Problem 1 is NP-hard, but a greedy algorithm achieves a 1/2-approximation ratio.

Qualitatively, given that each user does not benefit from having multiple copies of the same content available at different helpers, we expect that the solution of Problem 1 will try to make the largest number of popular contents available at each user and then it will maximize the hit probability over all the network.

B. Cooperative transmission

In this setup we assume base stations cooperation. We will use JT only for the base stations that already have the requested file cached and are in the user's neighborhood [1].

TABLE I
NOTATION SUMMARY

Notation	Description
M	size of a file
W	channel bandwidth
$\{1,\dots,U\}$	users
$\{1,\dots,H\}$	helpers
$\{1,\dots,F\}$	files
C	cache capacity
p_f	popularity distribution
g_{iu}	SNR from the helper i to user u
x_{if}	caching of the file f on the helper i
d_b	backhaul downloading time
$d_{r,m}(u)$	radio access delay for cache miss
$d_m(u)$	total delay for cache miss
$d_{r,h}(u,f)$	radio access delay for cache hit
$\bar{d}$	average delay in the network
$(c),(nc)$	*superscripts*, meaning cooperative and non-cooperative transmission

Cache miss: Upon a miss, the *radio access delay* and *backhaul delay* are the same as for the non-cooperative transmission and then the total delay is still d_m.

Cache hit: If user u requests file f, which is cached at least on one neighboring helper, the *backhaul delay* is 0 and all the helpers will coordinate their transmissions so that the SNRs sum at the mobile. The *radio access delay* is then:

$$d_{r,h}^{(c)}(u,f) = \frac{M}{W \log\left(1 + \sum_{h=1..H} x_{if} g_{iu}\right)}.$$

Optimization problem: The average delay is in this case

$$\bar{d}^{(c)} = \sum_{u,f} p_f \left(1_{k(u,f)=0} d_m(u) + 1_{k(u,f)>0} d_{r,h}^{(c)}(u,f)\right).$$

Carrying on calculations similar to those for the non-cooperative case we can conclude that

Problem 2 *(CoMP problem): In a cooperative setting minimizing the average delay is equivalent to solve the following maximization problem:*

maximize:

$$F^{(c)}(\mathbf{X}) = \sum_{u,f} p_f 1_{k(u,f)>0} \left(d_m(u) - d_{r,h}^{(c)}(u,f)\right) \quad (1)$$

subject to: $\sum_{f=1}^{F} x_{if} \leq C$, for $i = 1, \dots H$.

Likewise for the caching without cooperation, the higher hit probability means the lower downloading time. However, for the cooperative transmission, we can achieve higher radio access rates, comparing to the non-cooperative transmission, if there are possibilities to use joint transmission. Hence, to minimize the downloading delay of the files that can be served by multiple base stations, we should maximize the opportunities of the joint transmission, that we will call *CoMP opportunities*. In order to have a CoMP opportunity for a given

[1] Another possibility is to use also other base stations from the user's neighborhood, that do not necessary have the requested file cached. However in this case the base stations, that do not have the requested file cached, have to download it from the backhaul. As this would multiply the backhaul load by a factor equal to the number of BSs cooperating, we do not consider it as an option here.

user, the same file should be cached on several reachable base stations. On the other hand, the user will have higher hit probability if different files are cached. Hence the caching policy has two options: to increase the opportunities of the JT transmissions or to diversify the files.

Problem 2 is an integer programming problem and it is NP-hard, as it can be proven similarly to what done in [2].

In the next section we will show that the objective function is monotone and submodular, with constraints that can be written in matroid form. Based on this, we propose a greedy algorithm whose performance cannot be worse than $\frac{1}{2}$ of the optimal solution.

III. Optimal Content Placement for Uniform SNR

We first study CoMP problem when the SNRs between all the users and the reachable helpers are the same. i.e. $g_{iu} = g$ if the user u can connect to the helper i, or $g_{iu} = 0$ otherwise. In this case for any user u, $d_m(u) = d_m = d_b + \frac{M}{W\log(1+g)}$, and $d^{(c)}_{r,h}(u,f) = \frac{M}{W\log(1+k(u,f)g)}$. We can note, that $d^{(c)}_{r,h}(u,f)$ is now the function of $k(u,f)$: $d^{(c)}_{r,h}(u,f) = d^{(c)}_{r,h}(k(u,f))$. Problem 2 can then be formulated as follows:

Problem 2a *(Uniform SNR case)*:
maximize:

$$F^{(c)}(\mathbf{X}) = \sum_{u,f} p_f 1_{k(u,f)>0}\left[d_m - d^{(c)}_{r,h}(k(u,f))\right] \quad (2)$$

subject to:

$$\sum_{f=1..F} x_{if} \leq C, \text{ for } i = 1,\dots H \quad (3)$$

Let us define the ground set $S = \{s_{11},\dots s_{H1},\dots s_{1F},\dots s_{HF}\}$, where s_{if} is an abstract element denoting that helper i caches file f. Any matrix placement $\mathbf{X}$ can be put then in correspondence with a subset $X \subset S$, where s_{if} belongs to X if and only if $x_{if} = 1$, i.e. if content f is cached on helper i. We can then look at the function $F^{(c)}$ in (2) as a function of the set X rather than of the matrix $\mathbf{X}$. Similarly, constraint (3) defines the feasible sets X to be considered in the optimization problem. From now on we will then look at Problem 2a as the optimization of a set function.

A heuristic to solve Problem 2a is the following greedy algorithm. Start from an empty solution $T = \emptyset$, and then iteratively add to T the element $s \in S$ that does not violate the constraint (3) and maximizes $F^{(c)}(T \cup \{s\}) - F^{(c)}(T)$. We are going to prove that this algorithm achieves a 1/2-approximation ratio for Problem 2a. To this purpose we need the following two lemmas.

Lemma 1. *Constraints* (3) *define a partition matroid on the set S.*

This result was proven in original femto-caching paper [2, lemma 2], so we omit the proof.

Lemma 2. *The function* (2) *is monotone and submodular on the set S, if the backhaul delay is at least as large as the radio access delay:*

$$d_b \geq d^{(c)}_{r,h}(1). \quad (4)$$

Proof:

It is obvious that the function is monotone: when we add one more file to a helper we can only reduce the average delay and then increase the value of $F^{(c)}$. To check if the objective function is submodular, let us consider two sets X and X', $X \subset X' \subset S$. Let g_X be the gain of adding element s^*_{if} to the set X, that is:

$$g_X = F^{(c)}(X \cup \{s^*_{if}\}) - F^{(c)}(X).$$

Similarly the gain of adding the same element s^*_{if} to X' is $g_{X'}$.

For the users that do not see the helper i and for the files different from f nothing will change, so their corresponding terms will add up to 0. Let $U(i)$ be the set of users that are in the i helper's range. Let $k(u,f)$ and $k'(u,f)$ be the number of copies of the file f in the helpers of the user u respectively for placement X and placement X' before adding the element s^*_{if}. Then:

$$\begin{aligned} g_X - g_{X'} &= \sum_{u\in U(i)} p_f\left[d_m - d^{(c)}_{r,h}(k(u,f)+1)\right] \\ &- \sum_{u\in U(i)} p_f 1_{k(u,f)>0}\left[d_m - d^{(c)}_{r,h}(k(u,f))\right] \\ &- \sum_{u\in U(i)} p_f\left[d_m - d^{(c)}_{r,h}(k'(u,f)+1)\right] \\ &+ \sum_{u\in U(i)} p_f 1_{k'(u,f)>0}\left[d_m - d^{(c)}_{r,h}(k'(u,f))\right] \end{aligned}$$

For the users, for which $k(u,f) = k'(u,f)$, the gain is the same for both placements, so the difference becomes 0. Thus we need to look at the cases when $k(u,f) < k'(u,f)$.

To simplify the notation, let us divide the users $U(i)$ in 2 classes:

- U_1 are the users for which $k(u,f) > 0$.
  The gain for placement X: transition from downloading from k helpers to $k+1$ helpers.
  The gain for placement X': transition from downloading from k' helpers to $k'+1$ helpers.
- U_2 are the users for which $k(u,f) = 0$.
  The gain for placement X: transition from downloading through the backhaul to 1 helper.
  The gain for placement X': transition from downloading from k' helpers to $k'+1$ helpers.

$$\begin{aligned} &g_X - g_{X'} = \\ &\sum_{u\in U_1} p_f\left[d^{(c)}_{r,h}(k) - d^{(c)}_{r,h}(k+1) - (d^{(c)}_{r,h}(k') - d^{(c)}_{r,h}(k'+1))\right] \\ &+ \sum_{u\in U_2} p_f\left[(d_m - d^{(c)}_{r,h}(1) - (d^{(c)}_{r,h}(k') - d^{(c)}_{r,h}(k'+1))\right]. \end{aligned}$$

The first term is positive, because the function:

$$f(x) = d^{(c)}_{r,h}(x) - d^{(c)}_{r,h}(x+1),$$

is decreasing for $x > 0$. The following inequalities are true:

$$d_b \geq d_{r,h}^{(c)}(1) - d_{r,h}^{(c)}(2) \geq d_{r,h}^{(c)}(k') - d_{r,h}^{(c)}(k'+1). \quad (5)$$

The first inequality follows from the condition (4), the second again because the function $f(x)$ is decreasing. We can notice that $d_b = d_m - d_{r,h}^{(c)}(1)$, so the inequalities (5) guarantee that the second term is non-negative.

We showed that the gain of adding the element s_{fh}^* to the caching X is at least as big as adding it to the caching X'. Hence, the objective function (2) is submodular. ■

Since the objective function (2) is submodular over the set S and the constraints (3) define a partition matroid on this set, the described greedy algorithm achieves a 1/2-approximation for Problem 2a [17].

Before moving to the general problem, let us discuss the condition (4). Actually, from the inequalities (5), we see that the Lemma 2 holds for more relaxed constrains, when $d_b = d_m - d_{r,h}^{(c)}(1) \geq d_{r,h}^{(c)}(1) - d_{r,h}^{(c)}(2)$. This condition means that the performance gain from having one copy locally (as opposed to fetching the content over the backhaul) is greater than the additional gain from having two helpers caching and transmitting as opposed to just 1.

IV. Optimal Content Placement for Heterogeneous SNR

We will discuss now the general problem. Before formulating the lemma let us define two variables, first:

$$gain_1(u,h) = d_m(u) - \frac{M}{W\log(1+g_{hu})},$$

and second:

$$gain_2(u,h_1,h_2) = \frac{M}{W\log(1+g_{h_1u})} - \frac{M}{W\log(1+g_{h_1u}+g_{h_2u})} \quad (6)$$

Lemma 3. *The function* (1) *is monotone and submodular on the set S if for any user u and any two different helpers h_1 and h_2:*

$$gain_1(u,h_1) > gain_2(u,h_1,h_2). \quad (7)$$

We will not present the proof here as it is analogous to the same SNR case. Roughly speaking, the condition (7) means, that for a fixed user the SNRs, that it receives from the helpers to which it is allowed to connect, should not be very different. This can be done by putting the thresholds on the SNR. In general, the greedy algorithm can be applied even if the constraints of lemmas 2 and 3 do not hold, and it will probably find good enough solution. However, for some cases it can start to look for the solution in the wrong direction, thus we cannot guarantee a 1/2-approximation.

V. Simulation Results

We want to evaluate the improvement from solving the CoMP Problem 2 and then jointly optimizing caching and cooperative transmissions. As baseline, we consider the optimal caching allocation found in a non-cooperative scenario, i.e. the solution of the Femto Problem 1. In order to understand which part of the improvement is simply due to JT, we also consider a third scenario, where contents are allocated as determined by Problem 1, but the base stations take advantage of any opportunity to jointly transmit the content. We refer to this third scenario as *Femto+*.

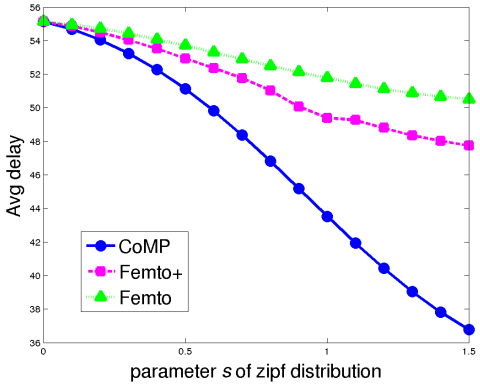

Fig. 1. Average delay in the network for uniform SNR evaluation, Sec. V-A

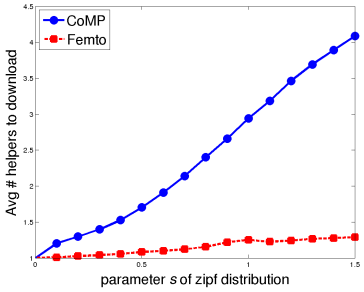

Fig. 2. Comparison of CoMP usage for the Femto and CoMP caching

Unless otherwise said, the setting considered in our simulations is the following. We have a squared area with side equal to 300m, where 25 helpers located at the centers of an hexagonal grid, so that the minimum distance between two helpers is 75m. The wireless channel has bandwidth $W = 5$ MHz, while the backhaul transmission rate is 100 Mbps. The cache of each helper can store 3 files out of a catalogue of 20 files, with size $M = 1$ Gbit. Content popularities follow a Zipf distribution with parameter s, i.e. $p_f \propto 1/f^s$. 100 users are placed uniformly at random in the area. Each user can connect to any helper less than $R = 100$ m away. We consider first the uniform SNR case.

A. Uniform SNR case evaluation

We consider here that the downlink SNR is 12 dB from any base station the user can reach.

Figure 1 compares the average time to download a file in Femto, Femto+ and CoMP cases, for different values of the parameter s of the Zipf distribution ($s = 0$ corresponds to the uniform distribution, the larger s the more skewed the distribution). First, we notice that Femto+ always outperforms Femto. This is expected because content allocation is exactly the same in the two cases, but in Femto+ case the base stations

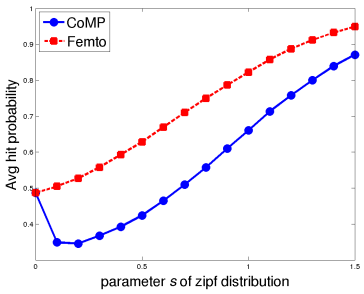

Fig. 3. Comparison of hit probability for the Femto and CoMP caching

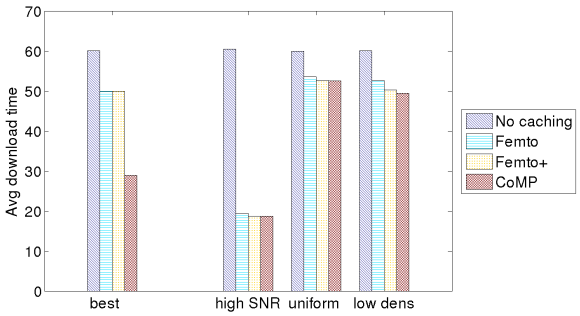

Fig. 4. CoMP caching performance evaluation

are allowed to cooperate, if they store the same file. While in the Femto case the user will always download from a single helper, in the Femto+ case he/she will download from the reachable helpers having a copy of the content.

Secondly, CoMP caching always outperforms femto-caching (both in the non-cooperative and cooperative cases). This is also expected because CoMP allocation is the solution of Problem 2 and then it minimizes the delay taking into account the possibility of cooperative transmissions. What is more interesting is the relative performance of the three approaches as s varies. When contents have the same popularity ($s \approx 0$), the improvement from storing an additional copy of the same file on one of the helpers is always smaller than the improvement from storing the first copy of a new content. In this case both Problem 1 and Problem 2 try to make available the largest number possible of files and then to maximize the hit rate. As s increases, popularities start becoming different and CoMP caching starts showing significant improvement comparing to femto-caching. For example, for moderately skewed distribution with $s = 0.6$ femto-caching achieves a 4.84% improvement in comparison to the Femto+ case and 6.52% in comparison to the Femto case. For a more skewed distribution with $s = 1.5$ the difference between the cachings is more significant: $22,97\%$ comparing to Femto+ case, $27,2\%$ comparing to Femto case. Note that the improvement cannot be simply explained through the possibility to exploit occasional opportunity for joint transmissions, otherwise the delay of CoMP and Femto+ would be much closer. The figure suggests that the two problems produce a very different content allocation.

This conclusion is supported by Figures 3 and 2 that show respectively i) the average number of helpers from which a user can download the content and ii) the hit probability. We observe that content allocations are different even for small values of s. CoMP on average stores in the caches more copies of the same content than Femto does, up to 3 times more for $s = 1.5$. Correspondingly, less contents are available to the user and the hit probability is smaller than for Femto and can even decrease in comparison to $s = 0$.

1) CoMP and Femto caching difference: We now move to study which parameters affect the performance gap between CoMP and Femto caching.

We start considering a reference scenario with $s = 1.5$ and $R = 300$ m, so that each user can potentially download from any helper. The corresponding average delay is indicated by the leftmost set of bars in Figure 4 denoted by "best." The bars show the different values for the three cases discussed above as well as a fourth setting when there is no cache. This scenario is particularly favorable to CoMP, that reduces the download time of 40% in comparison to Femto or Femto+. Let us try to understand why this setting gives such good results. In the experiments below we change this reference scenario one characteristic at a time.

Network links characteristics: In the reference scenario the largest component of the delay is due to the wireless link: indeed the file downloading time from a single helper is about 25s, while the backhaul delay is only 10s. In this situation, joint transmissions can lead to a significant reduction of the radio delay, so that it is convenient to increase the number of copies of the most popular contents, while paying the additional 10s to retrieve some less popular contents through the backhaul. In the second scenario we consider higher SNR (40dB) and slower backhaul rates (22 Mbps). Cache misses are now very penalized, so both CoMP and Femto caching will increase file diversity. The second set of bars in Figure 4 shows that the performance improvement is reduced to a few percent.

Popularity distribution: As we have already discussed above, when all the files are equally popular, the allocations produced by the Femto and the CoMP problems are the same. The third set of bars in Figure 4 shows that CoMP performs slightly better than Femto, but simply because of the occasional joint transmissions, as it is revealed by the fact that Femto+ achieves the same average delay.

Connectivity between the helpers and users: In the reference scenario each user can reach all the helpers. There are then many possibilities for base stations' cooperation. If we reduce network connectivity, CoMP has less opportunities to exploit. Moreover, in more cases users are reachable through a single helper and then both CoMP and Femto caches the most popular files at this helper. The last column of Figure 4 corresponds to an experiment where the transmission range is $R = 50$ m and then on average a user can reach 1.5 helpers. Again we see that CoMP and Femto cachings produce similar results.

Cache capacity: Reducing network connectivity can also be seen as reducing the total cache size a user can take advantage of. We can expect then that if, maintaining the

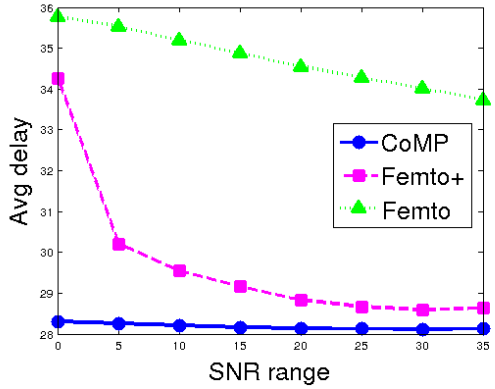

Fig. 5. CoMP caching performance evaluation

same connectivity, we reduce the size of each helper's cache, the performance gap decreases. Indeed, for both CoMP and Femto caching the greedy algorithm will first cache the most popular files, so that each user can download them from one of the reachable helpers. The first iterations of the greedy algorithm lead then to similar allocations in both cases. The allocations will rather differ at later iterations (if any): Femto caching will start to cache less popular files to increase file diversity, CoMP caching will make more copies of the popular files to create more CoMP opportunities. However, the smaller the caches, the smaller the number of iterations of the greedy algorithms and then the similar the allocations. The corresponding numerical results are not shown in the figure, but confirm this explanation.

B. Heterogeneous SNR case evaluation

Finally, in Figure 5 we present the results for the heterogeneous SNR case. In particular, SNR values for each helper-user pair have been drawn uniformly at random in the interval $[SNR_0 - \Delta SNR, SNR_0 + \Delta SNR]$. The figure shows the delay versus the SNR range ΔSNR when the average SNR value (SNR_0) is 17dB and $s = 1.5$. We see that also in this setting CoMP caching improves performance, but the improvement becomes smaller the more heterogeneous the SNR values. This is expected because the more different are the SNR values, the more important becomes the helper with the largest SNR value and the advantage of joint transmissions is reduced.

VI. Conclusions and Future Works

In this work we considered the problem of caching for the dense network, where the base stations can cooperate to transmit the same file to a user simultaneously. We formulated caching placement problem as an integer programming problem. First, we considered the case of uniform SNR. We showed under which conditions the problem can be solved efficiently with the greedy algorithm. Then we generalized the results for the initial problem. To evaluate the performance of the proposed CoMP caching, we conducted the simulations, where we compared this caching to the standard Femto caching and Femto caching, where the base stations are allowed to cooperate. The simulations allowed us to define the scenarios where CoMP caching performs significantly better than Femto caching.

For the future work we would like to consider also other CoMP techniques like scheduled beamforming, multiuser MIMO, etc.

References

[1] C. V. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update 2015-2020," *Cisco Public Information, February*, 2016.

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

[3] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Device-to-device collaboration through distributed storage," in *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE, 2012, pp. 2397–2402.

[4] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas, "Video delivery over heterogeneous cellular networks: Optimizing cost and performance," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1078–1086.

[5] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5g wireless networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2995–3007, 2016.

[6] E. Baştuğ, M. Bennis, and M. Debbah, "Social and spatial proactive caching for mobile data offloading," in *2014 IEEE international conference on communications workshops (ICC)*. IEEE, 2014, pp. 581–586.

[7] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, "Placing dynamic content in caches with small population," *arXiv preprint arXiv:1601.03926*, 2016.

[8] A. Giovanidis and A. Avranas, "Spatial multi-lru caching for wireless networks with coverage overlaps," *arXiv preprint arXiv:1602.07623*, 2016.

[9] 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.

[10] D. Lee, H. Seo, B. Clerckx, E. Hardouin, D. Mazzarese, S. Nagata, and K. Sayana, "Coordinated multipoint transmission and reception in lte-advanced: deployment scenarios and operational challenges," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, 2012.

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

[12] A. Liu and V. K. Lau, "Exploiting base station caching in mimo cellular networks: Opportunistic cooperation for video streaming," *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 57–69, 2015.

[13] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and impact of cloud computing on 5g signal processing: Flexible centralization through cloud-ran," *IEEE signal processing magazine*, vol. 31, no. 6, pp. 35–44, 2014.

[14] M. A. Maddah-Ali and U. Niesen, "Cache-aided interference channels," in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 809–813.

[15] W. C. Ao and K. Psounis, "Distributed caching and small cell cooperation for fast content delivery," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2015, pp. 127–136.

[16] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[17] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functionsi," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.