Caching at the Edge: Outage Probability

Estefanía Recayte, Andrea Munari
*Institute of Communications and Navigation of DLR (German Aerospace Center),
Wessling, Germany. Email: {estefania.recayte, andrea.munari}@dlr.de

***Abstract*—Caching at the edge of wireless networks is a key technology to reduce traffic in the backhaul link. However, a concentrated amount of requests during peak-periods may cause the outage of the system, meaning that the network is not able to serve the whole set of demands. The outage probability is a fundamental metric to take into account during the network design. In this paper, we derive the analytical expression of the outage probability as a function of the total amount of users requests, library size, requests distribution, cache size and capacity constraints on the backhaul resources. In particular, we focus on a scenario where end-users have no direct connection to the master node which holds the complete library of content that can be requested. A general formulation of the outage is derived and studied for two relevant caching schemes, i.e. the random caching scheme and the most popular caching schemes. The exact closed form expressions presented in this paper provide useful insights on how requests, memory and resources can be balanced when the parameters of a cache-enabled network have to designed.**

I. Introduction

The massive increase of multimedia content poses new challenges in wireless networks design. Typical approaches to counteract such enormous capacity demand consist in increasing spectral resources, i.e. bandwidth, or improving the spatial reuse, i.e. density of transmitters. However, in many cases these techniques may not be applicable due to their inherent costs or complexity. At the same time, sparing precious resources represents one of the most important objective for both satellite and terrestrial operators. A promising and feasible solution which is steadily gaining momentum both in the research and industry community consists in bringing the intended content at the edge of the network by means of caching [1]. Indeed, memorizing copies of content close to the users not only alleviates considerably the backhaul traffic, but may significantly reduce latency and power consumption. To achieve this goal, a two-step caching strategy is implemented, pre-fetching the content at the edge (e.g. at small base stations, relays or helpers) during network off-peak periods (*placement phase*) so as to serve the users without consuming backhaul capacity when the network is congested (*delivery phase*).

The effectiveness of caching is driven by a fundamental trade-off concerning the cost-related limits of physical cache. As a consequence, a proper balance between cache size and resource allocation has to be struck [2]. From this standpoint, a meaningful parameter for characterizing the system performance and which gives an insight of the design layout is given by the outage probability, i.e the probability that a user request cannot be served. Indeed, once the outage value at which the system should work is fixed then the memory size can be calculated based on the total available bandwidth and the total number of users.

Based on these considerations, several works have recently investigated outage in caching networks. Interesting results have been obtained in [3], computing the outage probability in device-to-device (D2D) cache enabled-networks where user can download the desired content from a one-hope neighbour. Instead, in [4] the outage probability of a user is given in a terrestrial network considering cache-enabled small base stations. In [5], [6] authors derive a closed form expression of the outage probability for a single user placed at the center of a dense small cell network. Instead in [7], authors studied optimization of caching schemes to improve cooperative communications in terms of outage performance gain in a scenario composed by multiple relays and a single user. Considering a multiple amplify-and-forward relay network, the content placement is optimized in [8] for reducing the outage when relays has unitary cache capacity and by considering a best relay selection.

This extensive body of research has provided a solid understanding of the potential of caching in serving the request of a specific user, assuming the existence of a connection to both local caches and to nodes keeping copy of all content of interest. On the other hand, scenarios in which multiple users attempt to retrieve content from the same cache, and cannot rely to a direct backhaul connection have not been tackled yet. Such setups are especially relevant in networks (e.g. beyond-5G systems and non-terrestrial networks (NTN)) which foresee a satellite component, employed to deliver content into local caches at ground base-stations. In this case, user terminals are typically not equipped with direct satellite connectivity, and the intermediate tier is responsible to forward content from one end to the other. In this context, only few works have investigated the performance of caching schemes [9], and the outage behaviour remains unexplored. Notably, new trade-offs arise, as a proper dimensioning of the satellite link capacity and cache size at ground relays becomes crucial in determining the quality of experience at the end-users.

To bridge this gap, we derive in this paper simple closed-form expressions for the outage probability, considering different statistics for file request distribution, a generic number of users and a capacity backhaul constraint. In particular, the performance of the network is analysed under two relevant

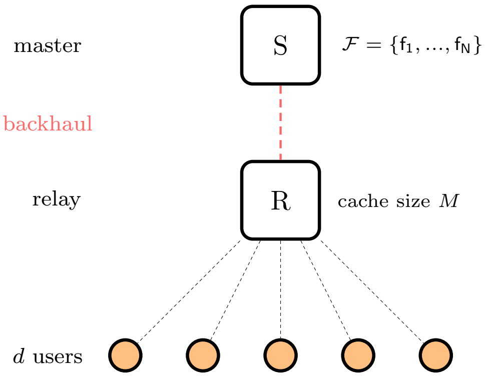
master
S
$\mathcal{F} = \{\mathsf{f}_1, ..., \mathsf{f}_\mathsf{N}\}$
backhaul
relay
R
cache size M
d users

Fig. 1. Reference system topology: d users/terminals are connected to the relay R with cache size M files, the relay is connected to the master node S through the backhaul link. S holds the whole library $\mathcal{F}$ of files.

caching schemes, i.e a random caching scheme and the most popular caching scheme. In the former case, we obtain exact expressions, whereas in the latter we overcome the problem complexity deriving a tight approximation of the outage expression which is validated via Monte-Carlo simulations. The presented formulations offer interesting insights, which are extensively discussed, and provide a useful design tool.

The rest of the paper is organized as follows. In Section II the system model is presented while in Section III the general expression of the outage is derived. In Section IV the outage for each scheme is studied. In Section V the numerical results are given. Finally, Section VI addresses the conclusions.

Notation

We use sans serif capital letters, e.g. X, for random variables (rvs) and their lower case counterparts, e.g. x, for their realizations. The probability mass function (pmf) of the rv X is denoted as P_X. Furthermore, we denote conditional pmfs as $\Pr\{\mathsf{X} = \mathsf{x} \,|\, Y = \mathsf{y}\} = p_\mathsf{X}(\mathsf{x}|\mathsf{y})$.

II. System Model

We consider a two-tier heterogeneous network where end-users are served by a cache-enabled node, which, in turn, is directly connected to a master node. While this setup applies to different network configurations, we will take as reference throughout our discussion the satellite topology illustrated in Fig. 1. Here, a satellite (S) holds a whole library $\mathcal{F} = \{\mathsf{f}_1, \cdots, \mathsf{f}_N\}$ of equal size files. On the ground, a cache-enabled relay (R) is connected via a backhaul link to S, and provides connectivity to users (or terminals) within its cell. Due to memory limitations, only a subset of $M \le N$ files can be stored by R. Moreover, as typical in current satellite-aided terrestrial networks, we assume that no direct link between users and S is available.

In such configuration, let d indicate the number of terminals that concurrently request content from the library, each independently picking a file to download. The requests are processed at R, which directly delivers files present in its cache, and retrieves via the backhaul link content which is not locally available. Aiming to characterise the trade-offs among memory size, backhaul dimensioning and content caching strategies, we assume that enough bandwidth is provided to correctly serve all users-to-relay connections, whereas a limited capacity is available on the relay-to-satellite link. Specifically, we denote the latter quantity by C, defined as the maximum number of *different* files that can be retrieved by R when attempting to serve users' requests.

Following this notation, the system is said to be in *outage* if the network cannot deliver content to all the d terminals, i.e., if the amount of content that has to be served through the backhaul link exceeds the capacity constraint C. It is worth noting that the event is driven not only by the available capacity, but also by how the relay caches files based on users demands. To explore this dimension, we consider two well-known and widely employed caching schemes, namely random placement (RaP) and most popular placement (MoP), which are explained next.

Random placement caching scheme (RaP)

In the RaP caching scheme, the request distribution is described as follows

$$p = \frac{1}{N},$$

i.e., each file belonging to $\mathcal{F}$ is assumed to be requested with the same probability p. Accordingly, during the placement phase R caches M files from the library uniformly at random, so that the probability for a requested file f_i to be present in cache is

$$\Pr\{\mathsf{f}_i \text{ is cached in RaP}\} = \frac{M}{N} \quad \forall i.$$

The RaP scheme represents a benchmark study case. The analysis of such approach is important, for instance, in scenarios where the actual file requests distribution is unknown.

Most popular placement caching scheme (MoP)

In the MoP caching scheme a file f_i is requested with probability p_i which follows a Zipf distribution [10] with shape parameter α such that

$$p_i = \frac{1}{\beta} i^{-\alpha} \qquad i = 1, ..., N \tag{1}$$

where $\beta := \sum_{j=1}^{N} j^{-\alpha}$ and the file-index i represents the order based on its popularity. During the placement phase, R caches the M most probable files of the library. The probability that file f_i is cached in the most popular scheme is then

$$\Pr\{\mathsf{f}_i \text{ is cached in MoP}\} = \begin{cases} 1 & i \le M \\ 0 & M < i \le N. \end{cases} \tag{2}$$

$$\mathsf{P}_{\mathsf{out}} = 1 - \left[\sum_{\mathsf{k}=0}^{C} \binom{d}{\mathsf{k}} \mathsf{P}_{\mathsf{nc}}^{\mathsf{k}} \,(1-\mathsf{P}_{\mathsf{nc}})^{d-\mathsf{k}} + \sum_{\mathsf{k}=C+1}^{d} \binom{d}{\mathsf{k}} \mathsf{P}_{\mathsf{nc}}^{\mathsf{k}} \,(1-\mathsf{P}_{\mathsf{nc}})^{d-\mathsf{k}} \sum_{\mathsf{z}=1}^{C} p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k}) \right]. \tag{7}$$

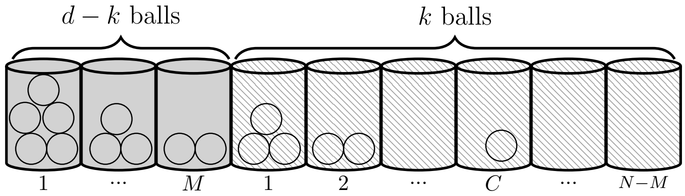

Fig. 2. Outage probability represented as the balls into bins problem. The M gray colored bins contain $d-\mathsf{k}$ requested files which are present in cache (subset $\mathcal{A}$). The $N-M$ dashed bins contain k requested files which are not present in cache (subset $\mathcal{B}$). The successful probability lies on calculating the probability of having at most C non empty bins in $\mathcal{B}$.

III. Outage Probability Formulation

To derive the outage behaviour of the system, we conveniently focus on the complementary quantity $\mathsf{P}_{\mathsf{succ}} = 1 - \mathsf{P}_{\mathsf{out}}$, capturing the probability for the network to succeed in serving all users' requests, i.e. that the amount of content to be retrieved via the backhaul does not exceed the capacity constraint C.[1]

More formally, let us indicate as Z the rv describing the number of different requested files which are not present in cache. The rv has alphabet $\{0, \cdots, \min(d, N-M)\}$, and allows to readily write

$$\mathsf{P}_{\mathsf{succ}} = \Pr\{\mathsf{Z} \le C\}. \tag{3}$$

Let us furthermore introduce the rv K, counting the number of *users* which have picked a file not present in cache. Note that the rv has alphabet $\{0, \dots, d\}$, and that $\mathsf{Z} \le \mathsf{K}$, since multiple users might ask for the same content. Leaning on this, the expression in (3) can be obtained via the law of total probability as

$$\mathsf{P}_{\mathsf{succ}} = \sum_{\mathsf{k}=0}^{d} \Pr\{\mathsf{Z} \le C \,|\, \mathsf{K} = \mathsf{k}\} p_{\mathsf{K}}(\mathsf{k}) \tag{4}$$

In turn, the summation in (4) can be split into two addends. Indeed, whenever K is lower than C, all the users can be served with success. Instead, when the number of terminals that request content not in cache is larger than the capacity constraint, the system succeeds only if the amount of distinct files requested does not exceed C (i.e., if two or more of such users have picked the same file). Applying these remarks, we then have

$$\mathsf{P}_{\mathsf{succ}} = \sum_{\mathsf{k}=0}^{C} p_{\mathsf{K}}(\mathsf{k}) + \sum_{\mathsf{k}=C+1}^{d} p_{\mathsf{K}}(\mathsf{k}) \sum_{\mathsf{z}=1}^{C} p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k}). \tag{5}$$

Let us now focus on K, and denote by P_{nc} the probability that a terminal selects a file not present in cache. Recalling that each user independently selects content, and that files are pre-fetched into the relay's cache, the r.v. follows a binomial distribution, i.e. $\mathsf{K} \sim \text{Bin}(d, \mathsf{P}_{\mathsf{nc}})$, and

$$p_{\mathsf{K}}(\mathsf{k}) = \binom{d}{\mathsf{k}} \mathsf{P}_{\mathsf{nc}}^{\mathsf{k}} \,(1-\mathsf{P}_{\mathsf{nc}})^{d-\mathsf{k}}. \tag{6}$$

Plugging (6) into (5) finally leads to the general expression for the outage probability reported in (7) at the top of the page. The formulation in (7) is handy, as it captures the behaviour of the system under a general caching strategy. In turn, P_{nc} and the conditional pmf $p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k})$ are specific to the implemented content storage policy, and will be derived in details in the next section for both the MoP and RaP approaches.

IV. Balls into bins problem applied to caching

In order to instantiate the calculation of the outage probability for the considered caching strategies, it is convenient to map our setting onto a balls into bins (BiB) setup. The general balls into bins (BiB) problem, see e.g. [11], consists in independently throwing d balls into N bins. As illustrated in Fig. 2, this can be cast to our case by having each bin associated to a file of the library, and by having balls represent user requests. Following this parallel, the possibility for more balls to land into the same bin corresponds to having multiple users asking for a common library element.

Without loss of generality, we split the bins into two subsets, labelled $\mathcal{A}$ and $\mathcal{B}$. The first has cardinality M, and indicates the files that are cached at R, while the second is composed of the $N-M$ bins that denote files only available via the backhaul link. Recalling the notation of Sec. III, a ball will then land into a bin of the two classes with probability $1-\mathsf{P}_{\mathsf{nc}}$ and P_{nc}, respectively, and the rv K counts the number of balls thrown onto bins in the second category. Furthermore, the pmf $p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k})$ can conveniently be seen as describing the number non-empty bins in $\mathcal{B}$ after d throws have been performed, conditioned on having K balls land into bins belonging to $\mathcal{B}$. Notably, as will be discussed in the following, exact closed-forms for such distribution can be derived when bins are picked uniformly, whereas tight approximations can be obtained when balls have different landing probabilities.

A. Random placement caching scheme

Following the BiB parallel, the RaP scheme corresponds to assuming that each ball is thrown uniformly at random over the available bins. The probability that exactly $\mathsf{Z} = \mathsf{z}$ bins out

[1] We implicitly restrict our attention to the only relevant case $d > C$. For $d \le C$, in fact, no outage occurs.

$$\mathsf{P}_{\mathsf{out}}^{(\mathsf{RaP})} = 1 - \left[\sum_{\mathsf{k}=0}^{C} \binom{d}{\mathsf{k}} \frac{M^{d-\mathsf{k}}}{N^d} (M-N)^{\mathsf{k}} + \sum_{\mathsf{k}=C+1}^{d} \binom{d}{\mathsf{k}} \sum_{\mathsf{z}=1}^{C} \frac{M^{d-\mathsf{k}}}{N^d} \binom{N-M}{\mathsf{z}} S(\mathsf{k},\mathsf{z})\,\mathsf{z}! \right]. \tag{10}$$

of the $N-M$ in class $\mathcal{B}$ are non empty given that $\mathsf{K}=\mathsf{k}$ launches have landed there can then be written as

$$p_{\mathsf{Z}}^{(\mathsf{RaP})}(\mathsf{z}|\mathsf{k}) = \frac{\binom{N-M}{\mathsf{z}} S(\mathsf{k},\mathsf{z})\,\mathsf{z}!}{(N-M)^{\mathsf{k}}} \tag{8}$$

where

$$S(\mathsf{k},\mathsf{z}) = \frac{1}{\mathsf{z}!} \sum_{j=0}^{\mathsf{z}} (-1)^j \binom{\mathsf{z}}{j} (\mathsf{z}-j)^{\mathsf{k}}.$$

The result follows by counting the favorable cases over all the possibles outcomes. In particular, we observe that there are $\binom{N-M}{\mathsf{z}}$ ways of choosing z files from $N-M$. For each such case, $S(\mathsf{k},\mathsf{z})$, denoting the Stirling number of the second kind [12], counts all the possible of ways in which k users can request for z different files. Finally, $\mathsf{z}!$ accounts for all the possible permutations of $S(\mathsf{k},\mathsf{z})$.

In order to compute the distribution of K, on the other hand, we observe that with the RaP policy a user requests a content that was not cached with probability

$$\mathsf{P}_{\mathsf{nc}}^{(\mathsf{RaP})} = 1 - \frac{M}{N} \tag{9}$$

so that, from (6),

$$p_{\mathsf{K}}^{(\mathsf{RaP})}(\mathsf{k}) = \binom{d}{\mathsf{k}} \left(1-\frac{M}{N}\right)^{\mathsf{k}} \left(\frac{M}{N}\right)^{d-\mathsf{k}}.$$

Leaning on these results, the outage probability $\mathsf{P}_{\mathsf{out}}^{(\mathsf{RAP})}$ can be derived by plugging (8) and (9) into (7), obtaining after some simple manipulations the final expression given in (10) at the top of the page.

B. Most popular placement caching scheme

In the MoP setup, files, i.e. bins, are chosen with different probability. To approach the problem, let us again condition our observation on having $\mathsf{K}=\mathsf{k}$ terminals selecting contents that are not present in cache. Under this assumption, we can focus on a simpler BiB problem, where k throws are performed, and each ball can fall solely onto one of the $N-M$ bins in $\mathcal{B}$. Specifically, recalling the Zipf distribution reported in (1), the $t-$th bin in this problem is chosen with probability

$$q_t = \frac{(M+t)^{-\alpha}}{\sum_{i=M+1}^{N} i^{-\alpha}} \qquad t = 1, ..., N-M$$

where $\sum_{t=1}^{N-M} q_t = 1$.

In this setup, the derivation of the probability to have $\mathsf{Z}=\mathsf{z}$ not empty bins, i.e., the sought $p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k})$, is known as the *occupancy problem*, for which, despite the simple conceptual formulation, a close-form solution is still elusive. To capture the performance of our system we thus recur to the approximation proposed in [11], and write the pmf of the number of non empty bins as

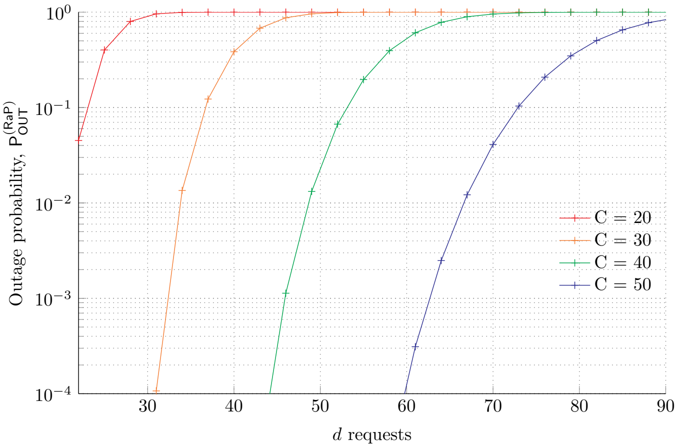

Fig. 3. Outage probability in the random placement scheme in function of the number of request d for different capacity levels $C = 20, 30, 40, 50$ when memory size $M = 10$, and library size $N = 100$.

$$p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k}) \approx \frac{1}{\sqrt{2\pi\sigma_{\mathsf{k}}^2}} \exp\left\{ -\frac{(\mathsf{z}-\mu_{\mathsf{k}})^2}{2\sigma_{\mathsf{k}}^2} \right\} \tag{11}$$

where

$$\mu_{\mathsf{k}} := (N-M) - \sum_{t=1}^{N-M} e^{-\mathsf{k}q_t}$$

$$\sigma_{\mathsf{k}}^2 := \sum_{t=1}^{N-M} e^{-\mathsf{k}q_t}\left(1-e^{-\mathsf{k}q_t}\right) - \frac{1}{\mathsf{k}}\left(\sum_{t=1}^{N-M} \mathsf{k}e^{-\mathsf{k}q_t} q_t\right)^2.$$

Referring to our caching problem, the equation given in (11) indicates the probability that k terminals request for z different files which are not cached.

Finally, we derive also for the MoP case the probability to request a non-cached file, i.e. $\mathsf{P}_{\mathsf{nc}}^{(\mathsf{MoP})}$. Leaning on (1) and recalling the caching policy in (2), we obtain

$$\begin{aligned} \mathsf{P}_{\mathsf{nc}}^{(\mathsf{MoP})} &= 1 - \sum_{i=1}^{M} p_i \\ &= 1 - \frac{1}{\beta} \sum_{i=1}^{M} \frac{1}{i^\alpha} = \frac{1}{\beta} \sum_{i=M+1}^{N} \frac{1}{i^\alpha}. \end{aligned} \tag{12}$$

From (12), the binomial pmf of K follows then as

$$p_{\mathsf{K}}^{(\mathsf{MoP})}(\mathsf{k}) = \frac{1}{\beta^d} \binom{d}{\mathsf{k}} \left(\sum_{i=M+1}^{N} \frac{1}{i^\alpha} \right)^{\mathsf{k}} \left(\sum_{i=1}^{M} \frac{1}{i^\alpha} \right)^{d-\mathsf{k}}$$

A good-approximated expression of outage probability $\mathsf{P}_{\mathsf{out}}^{(\mathsf{MoP})}$ in the most popular placement caching scheme is obtained by inserting (11) and (12) into (7). After simple manipulations we eventually obtain (13), reported at the top of next page.

$$\mathsf{P}_{\text{out}}^{(\text{MoP})} \approx 1 - \frac{1}{\beta^d}\left[\sum_{\mathsf{k}=0}^{C}\binom{d}{\mathsf{k}}\Big(\sum_{i=M+1}^{N}\frac{1}{i^{\alpha}}\Big)^{\mathsf{k}}\Big(\sum_{i=1}^{M}\frac{1}{i^{\alpha}}\Big)^{d-\mathsf{k}} + \sum_{\mathsf{k}=C+1}^{d}\binom{d}{\mathsf{k}}\sum_{\mathsf{z}=1}^{C}\frac{1}{\sqrt{2\pi\sigma_{\mathsf{k}}^2}}e^{-\frac{1}{2\sigma_{\mathsf{k}}^2}(\mathsf{z}-\mu_{\mathsf{k}})^2}\Big(\sum_{i=M+1}^{N}\frac{1}{i^{\alpha}}\Big)^{\mathsf{k}}\Big(\sum_{i=1}^{M}\frac{1}{i^{\alpha}}\Big)^{d-\mathsf{k}}\right]. \tag{13}$$

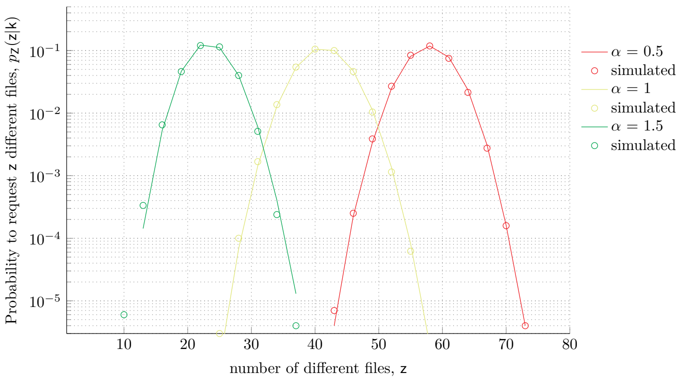

Fig. 4. Probability to request z different files given the demand of $\mathsf{k} = 100$ users when $N = 100$ and files follows the Zipf distribution with parameter $\alpha = 0.5, 1$ and 1.5. The solid curves represent our analytical approximation in (11) while circles are obtained via Monte-Carlo simulations.

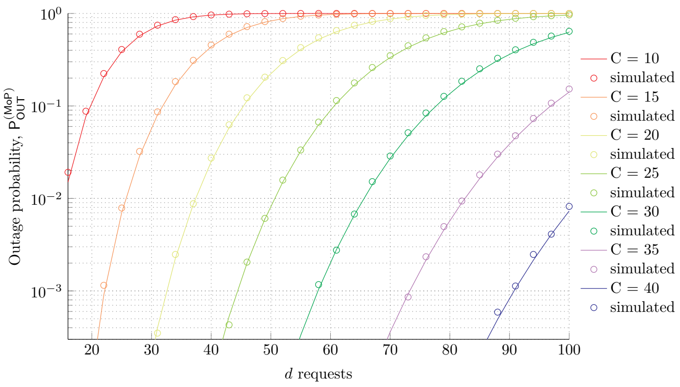

Fig. 5. Outage probability in the most popular caching scheme in function of the number of request d for different capacity levels $C = 10, 15, 20, 25, 30, 35, 40$ when $M = 10$, $\alpha = 1$ and $N = 100$. Given a constraint capacity C, the solid lines indicate the results obtained with our analytical approximation while dot markers indicate the corresponding result obtained by Monte-Carlo.

V. Results

In our first scenario, we assume a random placement in the cache. Users are connected to a relay with cache size $M = 10$ files, while the library cardinality is $N = 100$. In Fig. 3 the outage probability of RaP as a function of the number of requests for different values of backhaul capacity C is plotted. As expected, given d requests the outage probability decreases by increasing the backhaul capacity, since a larger number of requests can be served. However, this caching scheme requires high backhaul capacity for operating a relatively low levels of outage. For instance, the network demands a capacity $C = 40$ for ensure that simultaneously $d = 50$ requests are served with $\mathsf{P}_{\text{out}}^{(\text{RaP})} = 0.025$. The plot shows us that, in the case of equiprobable files, the benefit obtained from a cached network is minimal. As a matter of fact, this cache architecture does not significantly alleviate the traffic in the backhaul and, to operate with relatively low outage, the network needs to allocate backhaul resources in the order of the number of users that are active in the system.

Let us assume an special case of the MoP where $M = 0$, i.e. non of the files are cached and users request for content according to the Zipf distribution given in (1). Under this assumption, $p_{\mathsf{Z}}(\mathsf{z}|\mathsf{k})$ indicates the probability that k users demand for z different files. In Fig. 4 the probability that $\mathsf{k} = 100$ users request for z different files is plot for three different Zipf parameters, i.e. α, when the library size is $N = 100$. The solid curves represent the analytical approximation obtained in (11) while circles indicates the results obtained via Monte-Carlo. The plot shows the tightness and validity of the approximation for different values of α. When the skewness of the distribution is higher, i.e. $\alpha = 1.5$, user requests are concentrated in few files as shown by the green curve. In fact, for the considered Zipf parameter in mean $\mu_{100} = 23.36$ different files are requested. Instead, if we consider a lower Zipf parameter, for instance $\alpha = 0.5$, we observe that requests are spread over a larger number of files and in mean $\mu_{100} = 57.79$ different files z are requested.

In Fig. 5 the probability of outage as a function of the number of requests is plotted for the most popular placement caching scheme for different values of backhaul capacity. In the figure, solid lines report the analytical approximation, while markers the outcome of Montecarlo simulations. The reported trends were obtained assuming a shape parameter of the Zipf distribution $\alpha = 1$, (α typically assumes values in $[0.5, 1.5]$, see e.g. [10]). Moreover, the library size is $N = 100$ while the memory size $M = 10$, allowing a direct comparison with the RaP performance discussed earlier. As a first remark, we observe that the analytical results offer a very tight match to the simulations, prompting how the derived equations provide a simple yet effective tool for a preliminary system design. Furthermore, the plot shows the efficiency of the caching scheme due to the fact that more requests are concentrated in a small number of files. By increasing the backhaul capacity, significant gains in terms of number of requests served for a fixed outage probability is observed. For instance, a network operating at $\mathsf{P}_{\text{out}}^{(\text{MoP})} = 0.02$ can serve simultaneously $d = 27$ with only requiring a capacity $C = 15$. As soon as we double the backhaul capacity, i.e. $C = 30$, then $d = 68$ users can be served by ensuring the same outage probability. Unlike the RaP approach, the MoP caching scheme provides a huge gain

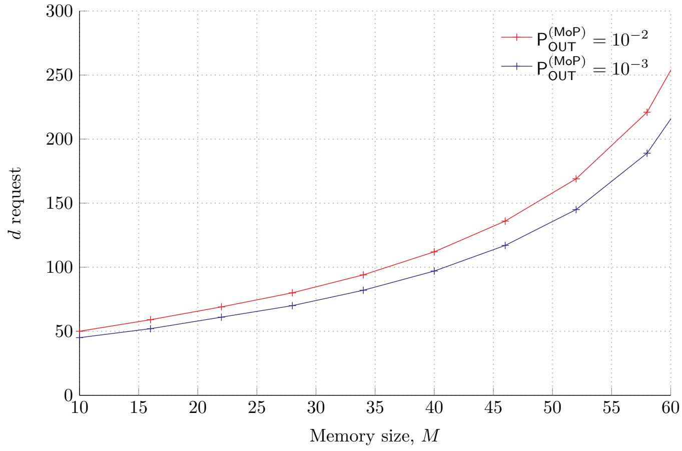

Fig. 6. Number of request that the system can support in the most popular placement scheme given in MoP $\mathsf{P}_{\mathsf{out}} = 10^{-2}$ and $\mathsf{P}_{\mathsf{out}} = 10^{-3}$ in function of the memory size when $C = 30, \alpha = 0.8, N = 100$.

of resources given that most of the requests are served by cached content.

The results obtained so far can be applied to design a cache network under fixed requirements. A relevant example could be when a operator has to decide the cache dimension given a constraint on the backhaul capacity while warranting a certain outage probability $\mathsf{P}_{\mathsf{out}}$. Based on (13), the number of users successfully served by the network can be derived as a function of the memory size given a $\mathsf{P}_{\mathsf{out}}$ of MoP. Thus, in Fig. 6 we show, for $\mathsf{P}_{\mathsf{out}}^{(\mathsf{MoP})} = 10^{-2}$ and for $\mathsf{P}_{\mathsf{out}}^{(\mathsf{MoP})} = 10^{-3}$, the number of users that can be served d as a function of the memory size M when the capacity $C = 30$, the library size $N = 100$ and the Zipf parameter $\alpha = 0.8$. The maximum number of users that can be simultaneously served can be determined from the plot by choosing a $\mathsf{P}_{\mathsf{out}}^{(\mathsf{MoP})}$ and fixing the memory size M.

Furthermore, we want to highlight that the results obtained for the MoP are also valid in a scenario where multiple relays are consider as long as relays have the same memory size. In fact it is easy to check that derivation does not change.

VI. Conclusions

In this work we consider an heterogeneous network with cache capability and we derive the outage probability when multiple users demand for content. In particular two caching schemes were considered. We derived a closed-form expression of the outage probability when a random caching scheme is on place. A well-approximated expression was obtained and then verified via Monte-Carlo for the most popular caching scheme. The outage probability was derived as a function of the number of total requests d, cache size M, total number of files N, requests distribution p (in case of RaP) or p_i (in case of MoP) and capacity constraint C. The results provide useful hints at the time of design a cached network. For example, one can have a quick and easy understand in trade-off between backhaul resources and maximum number of users that can be served.

References

[1] 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, 2014.

[2] 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, 2016.

[3] M. Ji, G. Caire, and A. F. Molisch, “The throughput-outage tradeoff of wireless one-hop caching networks,” *IEEE Transactions on Information Theory*, vol. 61, no. 12, pp. 6833–6859, 2015.

[4] E. Baştuğ, M. Bennis, and M. Debbah, “Cache-enabled small cell networks: Modeling and tradeoffs,” in *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, 2014, pp. 649–653.

[5] S. Tamoor-ul-Hassan, M. Bennis, P. H. J. Nardelli, and M. Latva-Aho, “Modeling and analysis of content caching in wireless small cell networks,” in *2015 International Symposium on Wireless Communication Systems (ISWCS)*, 2015, pp. 765–769.

[6] S. Tamoor-ul-Hassan, M. Bennis, P. H. J. Nardelli, and M. Latva-aho, “Caching in wireless small cell networks: A storage-bandwidth tradeoff,” *IEEE Communications Letters*, vol. 20, no. 6, pp. 1175–1178, 2016.

[7] G. Zheng, H. A. Suraweera, and I. Krikidis, “Optimization of hybrid cache placement for collaborative relaying,” *IEEE Communications Letters*, vol. 21, no. 2, pp. 442–445, 2017.

[8] L. Fan, N. Zhao, X. Lei, Q. Chen, N. Yang, and G. K. Karagiannidis, “Outage probability and optimal cache placement for multiple amplify-and-forward relay networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 373–12 378, 2018.

[9] E. Recayte, F. Lázaro, and G. Liva, “Caching in heterogeneous satellite networks with fountain codes,” *International Journal of Satellite Communications and Networking*, pp. 1–10, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1323

[10] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: evidence and implications,” in *IEEE Joint Conference of the IEEE Computer and Communications Societies (INFOCOM’99)*, 1999, pp. 126–134.

[11] S. K. Normal L Johnson, *Urn Models and Their Application*. New York: John Wiley & Sons, 1977, chapter 6.

[12] D. E. Knuth, *The art of computer programming*. United States: Addison-Wesley, 1969.