A Swiss Army Knife for Dynamic Caching in Small Cell Networks

Giovanni Neglia*, Emilio Leonardi‡, Guilherme Ricardo†*, and Thrasyvoulos Spyropoulos†

*Inria, Université Côte d’Azur, France, giovanni.neglia@inria.fr
‡Politecnico di Torino, Italy, leonardi@polito.it
†Eurecom, France, firstname.lastname@eurecom.fr

Abstract—We consider a dense cellular network, in which a limited-size cache is available at every base station (BS). Coordinating content allocation across the different caches can lead to significant performance gains, but is a difficult problem even when full information about the network and the request process is available. In this paper we present \(q\)LRU-\(\Delta\), a general-purpose dynamic caching policy that can be tailored to optimize different performance metrics also in presence of coordinated multipoint transmission techniques. The policy requires neither direct communication among BSs, nor a priori knowledge of content popularity and, under stationary request processes, has provable performance guarantees.

I. INTRODUCTION

In the last years, we have witnessed a dramatic shift of traffic at network edge, from the wired/fixed component to the wireless/mobile segment. This trend, mainly due to the huge success of mobile devices (smartphones, tablets) and their pervasive applications (Whatsapp, Instagram, Netflix, Spotify, Youtube, etc.), is expected to further strengthen in the next few years, as testified by several traffic forecasts. For example according to CISCO [1] in the 5 year interval ranging from 2017 to 2022 traffic demand on the cellular network will approximately increase by a factor of 9. As a consequence, the access (wireless and wired) infrastructure must be completely redesigned by densifying the cellular structure, and moving content closer to users. To this end, the massive deployment of caches within base stations of the cellular network is essential to effectively reduce the load on the back-haul links, as well as limit latency perceived by the user.

This work considers a dense cellular network scenario, where caches are placed at every Base Station (BS) and a significant fraction of users is “covered” by several BSs (whose cells are said to “overlap”). The BSs in the transmission range of a given user can coordinate to offer a seamless optimized caching service to the user and possibly exploit coordinated multipoint (CoMP) techniques [2] on the radio access. We remark that, as soon as there are overlapping BSs, finding the optimal static content allocation becomes an NP-hard problem, even when the request process is known, the metric to optimize is the simple cache hit ratio, and coordinated transmissions are not supported [3]. But realistic scenarios are more complex: popularities are dynamic and unknown a priori and more sophisticated metrics (e.g., PHY-based ones) that further couple nearby BSs-caches are of interest. Moreover, centralized coordination of hundreds or thousands of caches per \(\text{km}^2\) (e.g., in ultradense networks) is often infeasible or leads to excessive coordination overhead.

In such a context, our paper provides an answer to the open question about the existence of general (computationally efficient) distributed strategies for edge-cache coordination, which are able to provide some guarantees on global performance metrics (like hit ratio, retrieval time, load on the servers, etc.). In particular, we propose a new policy—\(q\)LRU-\(\Delta\)—which provably achieves a locally optimal configuration for general performance metrics.

\(q\)LRU-\(\Delta\) requires a simple modification to the basic behaviour of \(q\)LRU [4]. Upon a hit at a cache, \(q\)LRU-\(\Delta\) moves the corresponding content to the front of the queue with a probability that is proportional to the marginal utility of storing this copy. Upon a miss, it introduces the new content with some probability \(q\). \(q\)LRU-\(\Delta\) inherits from \(q\)LRU \(\mathcal{O}(1)\) computation time per request and memory requirements proportional to the cache size. Its request-driven operation does not need a priori knowledge of content popularities, removing a limit of most previous work. Some information about the local neighborhood (e.g., how many additional copies of the content are stored at close-by caches also serving that user) may be needed to compute the marginal gain. Such information, however, is limited, and can be piggybacked on existing messages the user sends to query such caches, or even on channel estimates messages mobile devices regularly send to nearby BSs [5]. As an
example, we show that $q$LRU-$\Delta$ is a practical solution to optimize hit ratio, retrieval time, load on the servers, etc., both when a single BS satisfies the user’s request and when multiple BSs coordinate their transmissions through CoMP techniques.

A. Related work

We limit ourselves to describe work that specifically addresses the caching problem in dense cellular networks.

The idea of coordinating the placement of contents at caches, which are closely located at BSs, was first proposed in [6] and its extension [3] under the name of FemtoCaching. This work assumes that requests follow the Independent Reference Model (IRM) and geographical popularity profiles are available, i.e. content requests are independent and request rates are known for all cell areas and their intersections. Finding the optimal content placement that maximizes the hit ratio is proved to be an NP-hard problem, but a greedy heuristic algorithm is shown to guarantee a $\frac{1}{2}$-approximation of the maximum hit ratio. In [7], the authors have generalized the approach of [6], [3], providing a formulation for the joint content-placement and user-association problem that maximizes the hit ratio. Efficient heuristic solutions have also been proposed. Authors of [8] have included the bandwidth costs in the formulation, and have proposed an on-line algorithm for the solution of the resulting problem. In [9], instead, the authors have designed a distributed algorithm based on Gibbs sampling, which is shown to asymptotically converge to the optimal allocation. [10] revisits the optimal content placement problem within a stochastic geometry framework and derives an elegant analytical characterization of the optimal policy and its performance. In [11] the authors have developed a few asynchronous distributed content placement algorithms with polynomial complexity and limited communication overhead (communication takes place only between overlapping cells), whose performance has been shown to be very good in most of the tested scenarios. Still, they assume that content popularities are perfectly known by the system. Moreover they focus on cache hit rates, and do not consider CoMP.

One of the first papers that jointly considers caching and CoMP techniques was [12]: two BSs storing the same file can coordinate its transmission to the mobile user in order to reduce the delay or to increase the throughput. The authors consider two caching heuristics: a randomized caching policy combined with maximum ratio transmission precoding and a threshold policy combined with zero forcing beamforming. While they derive the optimal parameter setting of such heuristics, these policies are in general suboptimal with no theoretical performance guarantee. [13] addresses this issue for joint transmissions techniques. The authors prove that delay minimization leads to a submodular maximization problem as long as the backhaul delay is larger than the transmission delay over the wireless channel. Under such condition, the greedy algorithm provides again a guaranteed approximation ratio. [14] considers two different CoMP techniques, i.e., joint transmission and parallel transmission, and derives formulas for the hit rate using tools from stochastic geometry.

Nevertheless, all aforementioned works hold the limiting assumption in [6] that geographical content popularity profiles are known by the system. Reliable popularity estimates over small geographical areas may be very hard to obtain [15]. On the contrary, policies like LRU and its variants (qLRU, 2LRU, . . .) do not rely on popularity estimation and are known to well behave under time-varying popularities. For this reason they are a de-facto standard in most of the deployed caching systems. [16] proposes a generalization of LRU to a dense cellular scenario. As above, a user at the intersection of multiple cells, can check the availability of the content at every covering cell and then download from one of them. The difference with respect to standard LRU is how cache states are updated. In particular, the authors of [16] consider two schemes: LRU-ONE and LRU-ALL. In LRU-ONE, each user is assigned to a reference cell/cache and only the state of her reference cache is updated upon a hit or a miss, independently from which cache the content has been retrieved from. In LRU-ALL, the state of all caches covering the user is updated.

Recently, [17] has proposed a novel approach to design coordinated caching polices in the framework of online linear optimization. A projected gradient method is used to tune the fraction of each content to be stored in a cache and regret guarantees are proved. Unfortunately, this solution requires to store pseudo-random linear combinations of original file chunks, and, even ignoring the additional cost of coding/decoding, it has $O(F)$ computation time per request as well as $O(F)$ memory requirements, where $F$ is the catalogue size. Also, coding excludes the possibility to exploit CoMP techniques, because all chunks are different.

Lastly, reference [18] proposes a novel approximate analytical approach to study systems of interacting caches, under different caching policies, whose predictions are surprisingly accurate. The framework builds upon the well known characteristic time approximation [19] for individual caches as well as an exponentialization approximation. We also rely on the same approx-


B. Paper Contribution

The main contribution of this paper is the proposal of qLRU-Δ, a general-purpose caching policy that can be tailored to optimize different performance metrics. The policy implicitly coordinates caching decisions across different caches also taking into account joint transmission opportunities. qLRU-Δ is presented in details in Sect. III after the introduction of our network model in Sect. II.

Sect. II is devoted to prove that, under a stationary request process, qLRU-Δ achieves a locally optimal configuration as the parameter q converges to 0. This means that it is not possible to replace a single content at one of the caches and still improve the performance metric of interest. The proof is technically sophisticated relying on the characterization of stochastically stable states through an opportune potential originally proposed in [20].

In order to illustrate the flexibility of qLRU-Δ, we show in Sect. V how to particularize the policy for two specific performance metrics, i.e., the hit rate and the retrieval delay under CoMP. While our theoretical guarantees hold only asymptotically, numerical results show that qLRU-Δ with q ∈ [0.01, 0.1] already approaches the performance of the static allocation obtained through greedy, which, while not provably optimal, is the best baseline we can compare to. Note that the greedy algorithm requires complete knowledge of network topology, transmission characteristics, and request process, while qLRU-Δ is a reactive policy that relies only on a noisy estimation of the marginal benefit deriving from a local copy.

We remark that the goal of qLRU-Δ and this paper is not to propose “the best” policy for any scenario with “coupled” caches, but rather a simple and easily customizable policy framework with provable theoretical properties. Currently, new caching policies designed for a particular scenario/metric are often compared with classic policies like LRU or LFU or the more recent LRU-ONE and LRU-ALL. This comparison appears to be quite unfair, given that these policies 1) ignore or only partially take into account the potential advantage of coordinated content allocations and 2) all target the hit-rate as performance metric. qLRU-Δ may be a valid reference point, while being simple to implement. A Swiss-army knife is a very helpful object to carry around, even if each of its tools may not be the best one to accomplish its specific task.

II. NETWORK MODEL

We consider a set of B base stations (BSs) arbitrarily located in a given region $R \subseteq \mathbb{R}^2$, each equipped with a local cache with size $C$. Users request contents from a finite catalogue of size $F$. A specific allocation of copies of content $f$ across the caches is specified by the vector $x_f = (x_{f1}, x_{f2}, \ldots, x_{fB})$, where $x_{fb} = 1$ (resp. $x_{fb} = 0$) indicates that a copy of $f$ is present (resp. absent) at BS $b$. Let $e^{(b)}$ be the vector with a 1 in position $b$ and all other components equal to 0. We write $x_f \oplus e^{(b)}$ to indicate a new cache configuration where a copy of content $f$ is added at base station $b$, if not already present. Similarly, $x_f \ominus e^{(b)}$ indicates a new allocation where there is no copy of content $f$ at $b$. Finally, we denote by $X_f(t) = (X_{f1}(t), \ldots, X_{fB}(t))$, the specific allocation at time $t$.

When user $u$ requests and receives content $f$, some network stakeholder achieves a gain that we assume to depend on user $u$, content $f$ and the current allocation of content $f$’s copies ($X_f(t)$). We denote the gain as $g_f(X_f(t), u)$. For example, if the key actor is the content server, $g_f(X_f(t), u)$ could be the indicator function denoting if $u$ can retrieve the content from one of the local caches (reducing the load on the server). If it is the network service provider, $g_f(X_f(t), u)$ could be the number of bytes caching prevents from traversing bottleneck links. Finally, if it is the user, $g_f(X_f(t), u)$ could be the delay reduction achieved through the local copies. We consider that $g_f(0, u) = 0$, i.e., if there is no copy of content $f$, the gain is zero.

The gain $g_f(X_f(t), u)$ may be a random variable. For example, it may depend on the instantaneous characteristics of the wireless channels, or on some user’s random choice like the BS from which the file will be downloaded. We assume that, conditionally on the network status $X_f(t)$ and the user $u$, these random variables are independent from one request to the other and are identically distributed with expected value $E[g_f(X_f(t), u)]$.

Our theoretical results hold under a stationary request process. In particular, we consider two settings. In the first one, there is a finite set of $U$ users located at specific positions. Each user $u$ requests the different contents according to independent Poisson process with rates $\Lambda_{f,u}$.
for $f \in \{1, 2, \ldots, F\}$. The total expected gain per time unit from a given placement $x_f$ is

$$G_f(x_f) = \sum_{u=1}^{U} \lambda_{f,u} \mathbb{E}[g_f(x_f, u)]. \quad (1)$$

In the second setting, a potentially unbounded number of users are spread over the region $R$ according to a Poisson point process with density $\mu(\cdot)$. Users are indistinguishable but for their position $r$. In particular, a user $u$ in $r$ generates a Poisson request process with rate $\lambda_f(r)$ and experiences a gain $g_f(x_f, r)$. The total expected gain from a given placement of content $f$ copies is in this case

$$G_f(x_f) = \int_{R} \lambda_f(r) \mathbb{E}[g_f(x_f, r)] \mu(r) dr. \quad (2)$$

In what follows, we will refer to the marginal gain from a copy at base station $b$. When the set of users is finite, we define the following quantities, respectively for a given user and for the whole network:

$$\Delta g_f^{(b)}(x_f, u) \triangleq g_f(x_f, u) - g_f(x_f \ominus e^{(b)}, u), \quad (3)$$

$$\Delta G_f^{(b)}(x_f) \triangleq G_f(x_f) - G_f(x_f \ominus e^{(b)}). \quad (4)$$

It is possible to define similarly $\Delta g_f^{(b)}(x_f, r)$ when users’ requests are characterized by a density over the region $R$. In what follows, we will usually refer to the case of a finite set of users, but all results hold in both scenarios.

Using (4) and the fact that $G_f(0) = 0$, it is easy to check that the gain from a given cache configuration $x_f = (x_f^{(1)}, x_f^{(2)}, \ldots, x_f^{(B)})$ can be computed as follows:

$$G_f(x_f) = \sum_{b=1}^{B} \Delta G_f^{(b)}(x_f^{(1)}, \ldots, x_f^{(B)}, 0, \ldots, 0). \quad (5)$$

We would like our dynamic policy to converge to a content placement that maximizes the total expected gain, i.e.,

$$\text{maximize} \quad G(x) \triangleq \sum_{f=1}^{F} G_f(x_f) \quad (6)$$

$$\text{subject to} \quad \sum_{f=1}^{F} x_f^{(b)} = C \quad \forall b = 1, \ldots, B,$$

$$x_f^{(b)} \in \{0, 1\} \quad \forall f = 1, \ldots, F,$$

$$\forall b = 1, \ldots, B.$$  

even in the absence of a priori knowledge about the request process. In the three specific examples we have mentioned above, solving problem (6) respectively corresponds to 1) maximize the hit ratio, 2) minimize the network traffic, and 3) minimize the retrieval time. This problem is in general NP-hard, even in the case of the simple hit ratio metric $\Delta$.

III. $q$LRU-$\Delta$

We describe here how our system operates and the specific caching policy we propose to approach the solution of Problem (6).

When user $u$ has a request for content $f$, it broadcasts an inquiry message to the set of BSs ($I_u$) it can communicate with. The subset ($J_{u,f}$) of those BSs that have the content $f$ stored locally declare their availability to user $u$. If no local copy is available, the user sends the request to one of the BSs in $I_u$, which will need to retrieve it from the content provider.$^1$ If a local copy is available ($J_{u,f} \neq \emptyset$) and only point-to-point transmissions are possible, the user sends an explicit request to download it to one of the BSs in $J_{u,f}$. Different user criteria can be defined to select the BS to download from (e.g., SNR, or pre-assigned priority list $\mathcal{L}$); for the sake of simplicity, in this paper, we assume that the user selects uniformly at random one of them. If CoMP techniques are supported, then all the BSs in $J_{u,f}$ coordinate to jointly transmit the content to the user.

Our policy $q$LRU-$\Delta$ works as follows. Each BS $b$ with a local copy ($b \in J_{u,f}$) moves the content to the front of the cache with probability proportional to the marginal gain due to the local copy, i.e.,

$$p_f^{(b)}(u) = \beta \Delta g_f^{(b)}(X_f(t), u), \quad (7)$$

where the constant $\beta$ guarantees that $p_f^{(b)}(u)$ is indeed a probability, i.e., it is between 0 and 1 and adimensional. At least one of the BSs without the content (i.e., those in $I_{u,f} \setminus J_{u,f}$) decides to store an additional copy of $f$ with probability

$$q_f^{(b)}(u) = q \delta \Delta g_f^{(b)}(X_f(t) \oplus e^{(b)}, u), \quad (8)$$

where $\delta$ plays the same role of $\beta$ above and $q$ is a dimensionless parameter in $(0, 1]$. We are going to prove that $q$LRU-$\Delta$ is asymptotically optimal when $q$ converges to 0. This result holds under different variants for the update rule at BSs without the content. For example, any number ($>0$) of BSs in $I_{u,f} \setminus J_{u,f}$ can randomly decide to retrieve or not the copy, and the probability $q_f^{(b)}(u)$ could be simply equal to a constant value $q$. We propose (8) because it is more likely to add copies that bring

1 This two-step procedure introduces some additional delay, but this is inevitable in any femtocaching scheme where the BSs need to coordinate to serve the content.
a large benefit $\Delta g_f^{(b)}(X_f(t) \oplus e^{(b)}, u)$. This choice likely improves convergence speed, and then the performance in non-stationary popularity environments. Moreover, as it will be clear from the discussion in the following section, our optimality result depends on $E[p_f^{(b)}(u)]$ being proportional to $E[\Delta g_f^{(b)}(X_f(t), u)]$. Then it is possible to replace $g_f^{(b)}(X_f(t), u)$ in (7) with any other unbiased estimator of $E[g_f^{(b)}(X_f(t), u)]$. We are going to show an example when this is useful in Sect. [V]

IV. OPTIMALITY OF qLRU-$\Delta$

A caching configuration $x = (x_1, x_2, \ldots, x_F)$ is locally optimal if it provides the highest gain $G(x) = \sum_{f=1}^{F} G_f(x_f)$ among all the caching configurations which can be obtained from $x$ by replacing one content in one of the caches.

Definition IV.1. A caching configuration $x$ is called locally optimal if, for any $y = x \oplus e^{(b)}$ or $y = x \ominus e^{(b)}$, it holds $G(x) \geq G(y)$.

We are going to prove that qLRU-$\Delta$ achieves a locally optimal configuration when $q$ vanishes. The result relies on two approximations: the usual characteristic time approximation (CTA) for caching policies (also known as Che’s approximation) [21], [19] and the new exponentialization approximation (EA) for networks of interacting caches originally proposed in [18]. The main results of this paper is the following:

Proposition IV.1. [loose statement] Under characteristic time and exponentialization approximations, a spatial network of qLRU-$\Delta$ caches asymptotically achieves a locally-optimal caching configuration when $q$ vanishes.

Before moving to the detailed proof, we provide some intuition about why this result holds. We observe that, as $q$ converges to 0, the cache exhibits two different dynamics with very different timescales: the insertion of new contents tends to happen more and more rarely ($q_f^{(b)}(u)$ converges to 0), while the frequency of position updates for files already in the cache is unchanged ($p_f^{(b)}(u)$ does not depend on $q$). A file $f$ at cache $b$ is moved to the front with a probability proportional to $\Delta g_f^{(b)}(X_f, u)$, i.e., proportional to how much the file contributes to improve the performance metric of interest. This is a very noisy signal: upon a given request, the file is moved to the front or not. At the same time, as $q$ converges to 0, more and more moves-to-the-front occur between any two file evictions. The expected number of moves-to-the-front file $f$ experiences is proportional to 1) how often it is requested ($\lambda_f(u)$) and 2) how likely it is to be moved to the front upon a request ($p_f^{(b)}(u)$). Overall, the expected number of moves is proportional to $\sum_{u} \lambda_f(u)E[\Delta g_f^{(b)}(X_f, u)]$, i.e., its contribution to the expected gain. By the law of large numbers, the random number of moves-to-the-front will be close to its expected value and it becomes likely that the least valuable file in the cache occupies the last position. We can then think that, when a new file is inserted in the cache, it will replace the file that contributes the least to the expected gain. qLRU-$\Delta$ then behaves as a greedy algorithm that, driven by the request process, upon insertions progressively replaces the least useful file from the cache, until it reaches a local maximum.

A. Characteristic Time Approximation

This is a now standard approximation for a cache in isolation, and one of the most effective approximate approaches for analysis of caching systems. CTA was first introduced (and analytically justified) in [21] and later rediscovered in [19]. It was originally proposed for LRU under the IRM request process, and it has been later extended to different caching policies and different requests processes [4], [22]. The characteristic time $T_c$ is the time a given content spends in the cache since its insertion until its eviction in absence of any request for it. In general, this time depends in a complex way from the dynamics of other contents requests. Instead, the CTA assumes that $T_c$ is a random variable independent from other contents dynamics and with an assigned distribution (the same for every content). This assumption makes it possible to decouple the dynamics of the different contents: upon a miss for content $f$, the content is retrieved and a timer with random value $T_c$ is generated. When the timer expires, the content is evicted from the cache. Cache policies differ in i) the distribution of $T_c$ and ii) what happens to the timer upon a hit. For example, $T_c$ is a constant under LRU, qLRU, 2LRU and FIFO and exponentially distributed under RANDOM. Upon a hit, the timer is renewed under LRU, qLRU and 2LRU, but not under FIFO or RANDOM. Under CTA, the instantaneous cache occupancy can violate the hard buffer constraint. The value of $T_c$ is obtained by imposing the expected occupancy to be equal to the buffer size. Despite its simplicity, CTA was shown to provide asymptotically exact predictions for a single LRU cache under IRM as the cache size grows large [21], [23], [24].

Once inserted in the cache, a given content $f$ will sojourn in the cache for a random amount of time $T_{S,f}$, independently from the dynamics of other contents.
\( T_{S,f} \) can be characterized for the different policies. In particular, if the timer is renewed upon a hit, we have:
\[
T_{S,f}^{(b)} = \sum_{k=1}^{M} Y_k + T_c^{(b)},
\]
where \( M \in \{0, 1, \ldots \} \) is the number of consecutive hits following a miss and \( Y_k \) is the time interval between the \( k \)-th hit and the previous content request.

We want to compute the expected value of \( T_{S,f}^{(b)} \) that we denote as \( 1/\nu_f^{(b)} \). When the number of users is finite, requests for content \( f \) from user \( u \) arrive according to a Poisson process with rate \( \lambda_{f,u} \). The time instants at which content \( f \) is moved to the front are generated by thinning this Poisson process with probability \( \beta \). The resulting sequence is then also a Poisson process with rate \( \lambda_{f,u} \). Finally, as request processes from different users are independent, the aggregate cache updates due to all users is a Poisson process with rate \( \beta U \).

The same result holds when we consider a density of requests over the region \( R \).

As the aggregate cache updates follow a Poisson process with rate \( \beta \Delta G_f^{(b)} \), \( \{Y_k\} \) are i.i.d. truncated exponential random variables with rate \( \beta \Delta G_f^{(b)} \) over the interval \( [0, T_c^{(b)}] \) and their expected value is
\[
\mathbb{E}[Y_k] = \frac{1}{\beta \Delta G_f^{(b)}} - \frac{T_c^{(b)}}{e^{\beta \Delta G_f^{(b)} T_c^{(b)}} - 1}.
\]
Moreover, the probability that no update occurs during a time interval of length \( T_c^{(b)} \) is \( e^{-\beta \Delta G_f^{(b)} T_c^{(b)}} \). Then \( M \) is distributed as a geometric random variable with values \( \{0, 1, \ldots\} \) and expected value
\[
\mathbb{E}[M] = \frac{1 - e^{-\beta \Delta G_f^{(b)} T_c^{(b)}}}{e^{-\beta \Delta G_f^{(b)} T_c^{(b)}}} = e^{\beta \Delta G_f^{(b)} T_c^{(b)}} - 1.
\]

We can then apply Wald Lemma to obtain:
\[

\nu_f^{(b)} \triangleq \frac{1}{\mathbb{E}[T_{S,f}^{(b)}]} = \frac{1}{\mathbb{E}[Y_k] \mathbb{E}[M] + T_c^{(b)}} = \frac{\beta \Delta G_f^{(b)}}{e^{\beta \Delta G_f^{(b)} T_c^{(b)}} - 1}.
\]

Here we simply write \( \Delta g_f^{(b)}(u) \) instead of \( \Delta g_f^{(b)}(X_f(t), u) \), because we are considering a single cache. Similarly, we write \( \Delta G_f^{(b)} \), instead of \( \Delta G_f^{(b)}(X_f(t)) \).

### B. Exponentialization Approximation

We consider now the case when \( B \) cells may overlap. The sojourn time of content \( f \) inserted at time \( t \) in cache \( b \) will now depend on the whole state vector \( X_f(\tau) \) for \( \tau \ge t \) (until the content is not evicted), because the content is updated with probability \( \beta \) depending on the marginal gain of the copy (and then on \( X_f(\tau) \)). EA consists to assume that the stochastic process \( X_f(t) \) is a continuous-time Markov chain. For each \( f \) and \( b \) the transition rate \( \nu_f^{(b)} \) from state \( X_f(t) = (x_f^{(b)}, y_f^{(b)}) \) to \( (x_f^{(b)} = 0, y_f^{(b)}) \) is given by \( \Delta G_f^{(b)} \cdot X_f(t) \). EA envisages to replace the original stochastic process, whose analysis appears prohibitive, with a (simpler) MC. \[13\] shows that this has no impact on any system metric that depends only on the stationary distribution in the following cases:

1) isolated caches,
2) caches using RANDOM policy,
3) caches using FIFO policy as far as the resulting Markov Chain \( X_f(t) \) is reversible.

Numerical results in \[13\] show that the approximation is practically very accurate also out of these specific cases.

### C. Transition rates of the continuous time Markov Chain as \( q \) vanishes

For a given content \( f \), let \( x_f \) and \( y_f \) be two possible states of the MC. We write \( x_f < y_f \) whenever \( x_f^{(b)} \le y_f^{(b)} \) for each \( b \) and there is at least one \( b_0 \) such that \( x_f^{(b_0)} < y_f^{(b_0)} \), and we say that \( y_f \) is an ancestor of \( x_f \), and \( x_f \) is a descendant of \( y_f \). Furthermore we denote by \( |x_f| = \sum_b x_f^{(b)} \) the number of copies of content \( f \) stored in state \( x_f \), and we call it the weight of the state \( x_f \). If \( x_f < y_f \) and \( |x_f| = |y_f| - 1 \), we say that \( y_f \) is a parent of \( x_f \) and \( x_f \) is a child of \( y_f \).

Now observe that by construction, transition rates in the MC are different from 0 only between pair of states \( x_f \) and \( y_f \), such that \( x_f < y_f \) or \( y_f < x_f \). The transition \( x_f \rightarrow y_f \) is called an upward transition, while \( y_f \rightarrow x_f \) is called a downward transition. A downward transition can only occur from a parent to a child (\( |x_f| = |y_f| - 1 \)). Let \( b_0 \) be the index such that \( x_f^{(b_0)} < y_f^{(b_0)} \). We have that the downward rate is
\[

\rho_{y_f \rightarrow x_f} = \nu_f^{(b_0)}(y_f) = \frac{\beta \Delta G_f^{(b_0)}}{e^{\beta \Delta G_f^{(b_0)}(y_f) T_c^{(b_0)}} - 1}.
\]

Upward transitions can occur to states that are ancestors. The exact transition rate between state \( x_f \) and state \( y_f \) with \( x_f < y_f \) can have a quite complex expression, because it depends on the joint decisions of the BSs in
Lemma IV.2. Follows that a downward transition from a parent $x_f \to y_f$ occurs, if $|y_f| - |x_f|$ BSs independently store, each with probability proportional to $q$, an additional copy of the content $f$ in their local cache. It follows that:

$$\rho_{|x_f\to y_f|} \propto q^{y_f^\perp|x_f^\perp}.$$  

(12)

Now, as $q \to 0$ for every $f$ every upward rate $r_{x_f\to y_f}$ tends to 0. Therefore, the characteristic time of every cell $T_{\delta}^{(b)}(q)$ must diverge. In fact, if it were not the case for a cache $b$, none of the contents would be found in this cache asymptotically, because upward rates tend to zero, while downward rates would not. This would contradict the set of constraints:

$$\sum_f \sum_{x_f} x_f^{(b)} \pi(x_f) = C, \quad \forall b$$

imposed by the CTA, where $\pi(\cdot)$ is the MC stationary distribution. Therefore necessarily $T_{\delta}^{(b)}(q) \to \infty$ for every cell $b$. More precisely, we must have $T_{\delta}^{(b)}(q) = \Theta(\log 1/q)$ at every cache, otherwise we fail to meet (13). There exist then positive constants $a_i^{(b)}(q)$ and $a_u^{(b)}(q)$, such that $T_{\delta}^{(b)}(q)/\log(1/q)$ asymptotically belongs to $[a_i^{(b)}, a_u^{(b)}]$. We would like to characterize more precisely the growth rate of $T_{\delta}^{(b)}(q)$, i.e., to show that $T_{\delta}^{(b)}(q) \propto \log 1/q$ but our attempts have been unfruitful. Nevertheless, we can prove that there exists a sequence $\{q_n\}_n$ converging to 0 and a set of positive constants $\{\gamma_b, b = 1, \ldots, B\}$ such that $T_{\delta}^{(b)}(q_n) \sim \frac{1}{\beta \gamma_b} \log \frac{1}{q_n}$ for all $b \in \{1, \ldots, B\}$. In fact, consider any sequence $\{q_n\}_n$ converging to zero, the sequence of vectors $(T_{\delta}^{(b)}(q_n)/\log(1/q_n))_{b \in \{1, \ldots, B\}}$ belongs to the compact interval $\prod_{b=1}^B [a_i^{(b)}, a_u^{(b)}]$ and then admits a converging subsequence because of Bolzano-Weierstrass theorem. The limit of such subsequence is the vector $(\beta \gamma_1, \ldots, \beta \gamma_B)$. From this result and (11), it follows that a downward transition from a parent $y_f$ to a child $x_f = y_f \oplus e^{(b_0)}$ occurs with rate

$$\rho_{y_f \to x_f} \propto q_n \Delta G_f^{(b_0)}(y_f)/\gamma_{b_0}.$$  

The following lemma summarises the results of this section.

**Lemma IV.2.** Consider two neighbouring states $x_f$ and $y_f$ with $x_f < y_f$. There exists a sequence $\{q_n\}_n$ converging to zero and a set of positive constants $\{\gamma_b, b = 1, \ldots, B\}$, such that

$$\rho_{x_f \to y_f} \propto q_n^{y_f^\perp|x_f^\perp},$$

if $x_f = y_f \oplus e^{(b_0)}$, then

$$\rho_{y_f \to x_f} \propto q_n \Delta G_f^{(b_0)}(y_f)/\gamma_{b_0}.$$  

**D. Stochastically stable states**

In this sub-section, we first introduce the key concept of stochastically stable state(s), in which, as $q$ converges to 0, the system gets trapped. Then, we provide a preliminary characterization of stochastically stable states (Lemma IV.4), which will be useful in subsection IV-F to prove that they correspond necessarily to locally optimal configurations.

Let us consider the uniformization of the continuous time MC $X_f(t)$ with an arbitrarily high rate. We denote with $X_f(k)$ the so obtained discrete time MC, whose transition probability matrix is $P_f,q$. For $q = 0$, the set of contents in the cache does not change, each state is an absorbing one and any probability distribution is a stationary probability distribution for $P_f,0$. We are rather interested in the asymptotic behaviour of the MC when $q$ converges to 0. For $q > 0$ the MC is finite, irreducible and aperiodic and then admits a unique stationary probability $\pi_f,q$.

**Definition IV.2.** A state $x_f$ is called stochastically stable if $\lim_{q \to 0} \pi_f,q(x_f) > 0$.

We are going to characterize such states. For what we have said above, the probability to move from $x$ to an ancestor (if not zero) is $P_{f,q}(x_f,y_f) \propto q^{y_f^\perp|x_f^\perp}$. The probability to move from $y_f$ to the child $x_f$ without the copy in $b_0$ is $P_{f,q}(y_f,x_f) \propto q^{G_f^{(b_0)}(y_f)/\gamma_{b_0}}$. For each possible transition, we define its direct resistance to be the exponent of the parameter $q$, then $r(x_f,y_f) = |y_f^\perp|x_f^\perp|$, $r(y_f,x_f) = \Delta G_f^{(b_0)}(x_f)/\gamma_{b_0}$ and $r(x_f,x_f) = 0$. Observe that the higher the resistance, the less likely the corresponding transition. We consider a weighted graph $G_f$, whose nodes are the possible states $x_f \in \{0,1\}^B$ and edges indicate possible direct transitions and have a weight equal to the corresponding resistance. Given a sequence of transitions $(x_1^f, x_2^f, \ldots, x_j^f)$ from state $x_1^f$ to state $x_j^f$ (or equivalently a path in $G_f$), we

---

3 For simplicity, we still refer to $q$ converging to 0, but the results hold for the vanishing sequence $\{q_n\}_n$ introduced above.

4 This is guaranteed if insertion probabilities in $[0,1]$ are positive. In some specific settings, it may be $\Delta G_f^{(b)}(X_f(t) \oplus e^{(b)}, u) = 0$ for each $u$. We can then consider $q_f^{(b)}(u) = q_f \max(\Delta G_f^{(b)}(X_f(t) \oplus e^{(b)}, u), \epsilon)$ with $\epsilon > 0$, or simply $q_f^{(b)}(u) = q_f$. 

define its resistance to be the sum of the resistances, i.e., \( r(x_i, x_j^2, \ldots, x_j^n) = \sum_{i=1}^{n-1} r(x_i^j, x_j^{j+1}) \).

We define the potential of state \( x_f \) (\( V_f(x_f) \)) as the resistance of the minimum weight in-tree (or anti-arborescence) in \( G_f \) rooted to \( x_f \). Intuitively, the potential is a measure of the general difficulty to reach state \( x_f \) from all other nodes.

The following lemma formalizes the intuition that the minimum weight in-tree will not include an edge, if it is possible to include the corresponding nodes as well as a third one at the same cost.

**Lemma IV.3.** Consider a strongly connected weighted directed graph \( G = (V, E) \) with weight \( w(u, v) > 0 \) for each edge \((u, v) \in E\). A minimum cost spanning tree rooted in a given node \( r \) does not contain edges \((u, v) \in E\), such that there exists a node \( t \) for which \((u, t) \) and \((t, v) \) belong to \( E \) and \( w(u, v) = w(u, t) + w(t, v) \).

**Proof.** Let us denote by \( T \) a minimum cost spanning tree rooted in \( r \). We prove the result by contradiction. Assume the edge \((u, v) \) with the property indicated above belongs to \( T \). Let us now consider the set of edges \( S = \{T \setminus \{(u, v)\}) \cup \{(u, t), (t, v)\} \). Any node can reach node \( r \) through the edges in \( S \). Note that \((u, t) \) cannot belong to \( T \), because any node can only have an outgoing edge and \((u, v) \) belongs to \( T \). If \( |S| = |T| \) (i.e., if \((t, v) \in T\) ), then \( S \) is also an in-tree rooted in \( r \) and its weight is strictly smaller than the weight of \( T \). If \( |S| = |T| + 1 \) (i.e., if the path from \( t \) to \( r \) in \( T \) does not use the edge \((t, v) \) ), then \( S \) is not an in-tree and the sum of the weights of the edge in \( S \) and in \( T \) is equal. It is then possible to remove an edge from \( S \) to obtain an in-tree with strictly smaller weight.

We can now characterize the set of stochastically stable sets:

**Lemma IV.4.** A state \( x_f \) is stochastically stable if and only if its potential \( V(x_f) \) is minimal.

**Proof.** The family of Markov chains \( \{P_{f, q}\} \) is a regular perturbation [20] properties (6-8) and then the result follows from Theorem 4 of [20], which provides an analogous characterization. Our analysis simplifies the one in [20]. In fact, the analysis in [20] focuses on the complete graph \( G_f \), whose vertices are the recurrent communication classes of \( P_{f, 0} \) and the weight of the edge between class \( C_i \) and class \( C_j \) (\( R(C_i, C_j) \)) is the minimum resistance of all the possible paths between any state \( x_f \in C_i \) and \( x_f' \in C_j \). The potential of class \( C_i \) is then the minimum weight in-tree in \( G_f \). In our case, each communication class includes a single state, therefore \( G_f \) and \( G_f' \) have the same set of nodes. \( G_f' \) has more edges (it is a complete graph), but one can show that minimum weight in-trees in \( G_f' \) include only edges that correspond to direct transitions between the two associated state, i.e., edges that also appear in \( G \) (Lemma IV.3).

For each content \( f \) we are then able to characterize which configurations are stochastically stable as \( q \) converges to 0. Moreover, this set of configurations must satisfy the constraint (13) at each base station \( b \). We conclude this section introducing the concept of *jointly stochastically stable* cache configurations.

**Definition IV.3.** A cache configuration \( x = (x_1, x_2, \ldots, x_F) \) is jointly stochastically stable if 1) for each content \( f \) \( x_f \) is stochastically stable, 2) \( x \) satisfies (13) for each \( b \).

**E. Dominant transitions**

In the proof or Proposition IV.1 the concept of dominant transitions will be used.

**Definition IV.4.** Given two neighboring states \( x_f \) and \( x_f' \), we say that the transition \( x_f \rightarrow x_f' \) is dominant if \( r(x_f, x_f') \leq r(x_f', x_f) \).

Let \( y_f \) be a parent of \( x_f \) with \( x_f^{(b)} = 0 \) and \( y_f^{(b)} = 1 \), we observe that the upward transition \( x_f \rightarrow y_f \) is dominant if and only if \( \Delta G_f^{(b)}(y_f) \geq \gamma_b \). Similarly the downward transition \( y_f \rightarrow x_f \) is dominant if and only if \( \Delta G_f^{(b)}(y_f) \leq \gamma_b \). Note that both transitions \( x_f \rightarrow y_f \) and \( y_f \rightarrow x_f \) are dominant if \( \Delta G_f^{(b)}(y_f) = \gamma_b \).

Let us also consider the function of state \( x_f \)

\[
\phi(x_f) \triangleq G_f(x_f) - \sum_{b \mid x_f^{(b)}=1} \gamma_b. \tag{14}
\]

The following lemma guarantees that the function \( \phi(.) \) decreases along a non-dominant transition.

**Lemma IV.5.** The transition \( x_f \rightarrow y_f \) is dominant if and only if \( \phi(y) \geq \phi(x) \).

**Proof.** Let \( b' \) be the index at which \( x_f \) and \( y_f \) differ. By definition, the upward transition \( x_f \rightarrow y_f \) is dominant if and only if \( \Delta G_f^{(b')} (y_f) \geq \gamma_b \). This inequality is equivalent to:

\[
\phi(y_f) = G_f(y_f) - \sum_{b \mid y_f^{(b)}=1} \gamma_b
\]
\[
= G_f(x_f) - \sum_{b \mid x_f^{(b)}=1} \gamma_b + \Delta G_f^{(b')} (y_f) - \gamma_b
\]
\[
\geq \phi(x_f).
\]

The proof when \( x_f \rightarrow y_f \) is a downward dominant transition is similar.
F. Optimality proof

Now we can state formally our result.

Proposition IV.1 Under characteristic time and exponentialization approximations, let \( \{q_0\}_n \) and \( \{\gamma_b, b = 1, \ldots, B\} \) be the sequence and the constants in Lemma IV.2. Consider the spatial network of qLRU-\( \Delta \) caches, where cache \( b \) selects the parameter \( q_0^{(b)} = q_0^{(b)} b \). As \( n \) diverges, a jointly stochastically stable cache configuration is also locally-optimal.

Proof: Given a cache configuration \( x = (x_1, x_2, \ldots, x_F) \), let \( G(x) \) denote the global gain across all contents, i.e., \( G(x) = \sum_{f=1}^{F} G_f(x_f) \).

A jointly stochastically stable cache configuration \( x \) is locally optimal, if and only if changing one content at a given cache does not increase the global gain \( G(x) \). Without loss of generality, we consider to replace content \( f_1 \) present at cache \( B \) with content \( f_2 \). Then, the cache allocation \( x \) changes from \( x_{f_1} = (x_{f_1}^{(B)} = 1, x_{f_1}^{(-B)}) \) and \( x_{f_2} = (x_{f_2}^{(B)} = 0, x_{f_2}^{(-B)}) \) to a new one cache allocation \( x' \), such that \( x'_{f_1} = (x'_{f_1}^{(B)} = 0, x'_{f_1}^{(-B)}) \) and \( x'_{f_2} = (x'_{f_2}^{(B)} = 1, x'_{f_2}^{(-B)}) \). From (5), we obtain that

\[
G(x) \geq G(x') \iff G_f(x_{f_1}) + G_{f_2}(x_{f_2}) \geq G_f(x'_{f_1}) + G_{f_2}(x'_{f_2}) \iff \Delta G^{(B)}_{f_1}(x_{f_1}) \geq \Delta G^{(B)}_{f_2}(x'_{f_2}).
\]

In order to prove \( \Delta G^{(B)}_{f_1}(x_{f_1}) \geq \Delta G^{(B)}_{f_2}(x'_{f_2}) \), we will show that

\[
\Delta G^{(B)}_{f_1}(x_{f_1}) \geq \gamma_B \quad (16) \quad \Delta G^{(B)}_{f_2}(x'_{f_2}) \leq \gamma_B. \quad (17)
\]

We start observing that we can ignore all upward transitions with resistance larger than 1, i.e., all transitions to ancestors that are not also parents. In fact, by applying multiple times Lemma IV.3, we can conclude that none of them is used in the minimum weight in-tree that defines the potential. Note how the flexibility in defining the qLRU-\( \Delta \) update rule upon misses comes from the fact that 1) resistances only depend on the exponent of \( q \) (and then additional positive factors like those appearing in (5) do not play a role), 2) upward edges with resistance larger than 1 do not contribute to the potential (and then it does not matter how many caches retrieve a copy of the content as far as at least one of them does with some probability that scales as \( q \)).

The state \( x'_{f_1} \) is a child of \( x_{f_1} \), then \( r(x_{f_1}, x'_{f_1}) = \Delta G^{(B)}_{f_1}(x'_{f_1})/\gamma_B \). Consider the in-tree \( T \) rooted in \( x_{f_1} \) with minimal resistance and let \( R(T) = V(x_{f_1}) \) denote its resistance and \( (x_1^{f_1} = x'_{f_1}, x_2^{f_1}, \ldots, x_B^{f_1} = x_{f_1}) \) be the sequence of transitions in \( T \) from \( x'_{f_1} \) to \( x_{f_1} \). One of these transitions, say \( x'_{f_1} \to x_{f_1}^{(i+1)} \) corresponds to store the content \( f_1 \) in the cache \( B \) and has resistance \( 1 \). Consider now the in-tree \( T' \) rooted in \( x'_{f_1} \), obtained from \( T \) removing the edge \( (x_{f_1}^{(i)}), x_{f_1}^{(i+1)} \) and adding the edge \( (x_{f_1}, x'_{f_1}) \). Its resistance is \( R(T') = R(T) - 1 + \Delta G^{(B)}_{f_1}(x_{f_1})/\gamma_B \). From \( R(T') \geq V(x'_{f_1}) \geq V(x_{f_1}) = R(T) \) it follows (16). A sketch of this construction is in Fig. 1a.

The proof of (17) is slightly more complex. First we prove it under the assumption that \( \gamma_b = \gamma \) for every cell \( b \), then we consider the general case. We prove (17) by contradiction. Let us assume that \( \Delta G^{(B)}_{f_2}(x'_{f_2}) > \gamma_B \).

In such case \( x'_{f_2} \to x_{f_2} \) is not a dominant (downward) transition, and \( \phi(x_{f_2}) < \phi(x'_{f_2}) \).

Let now \( T \) denote the in-tree rooted in \( x_{f_2} \) with minimal resistance and \( \mathcal{P} = (x_{f_2}^1, x_{f_2}^2, \ldots, x_{f_2}^B = x_{f_2}) \) be the sequence of transitions in \( T \) from \( x_{f_2}^1 \) to \( x_{f_2} \). At each transition \( x_{f_2}^{i-1} \to x_{f_2}^i \) only one state variable changes, we denote by \( b_i \) the corresponding index, representing the base station at/from which a copy of content \( f_2 \) is added/removed. By construction we have:

\[
\sum_{1 \leq l \leq k} \phi(x_{f_2}^l) - \phi(x_{f_2}^{l-1}) = \phi(x_{f_2}) - \phi(x'_{f_2}) < 0
\]

If transition \( x_{f_2}^{l-1} \to x_{f_2}^l \) is upward:

\[
\phi(x_{f_2}^l) - \phi(x_{f_2}^{l-1}) = \Delta G^{(b_i)}_{f_2}(x_{f_2}^l) = \gamma_{b_i}
\]

\[
\gamma_{b_i}[r(x_{f_2}^l, x_{f_2}^{l-1}) - r(x_{f_2}^{l-1}, x_{f_2}^l)] . \quad (18)
\]

If transition \( x_{f_2}^l \to x_{f_2}^{l+1} \) is downward,

\[
\phi(x_{f_2}^l) - \phi(x_{f_2}^{l-1}) = \gamma_{b_i} - \Delta G^{(b_i)}_{f_2}(x_{f_2}^{l-1})
\]

\[
\gamma_{b_i}[r(x_{f_2}^l, x_{f_2}^{l-1}) - r(x_{f_2}^{l-1}, x_{f_2}^l)] , \quad (19)
\]

obtaining the same expression in terms of the resistances. Therefore

\[
\sum_{1 \leq l \leq k} \gamma_{b_i}[r(x_{f_2}^l, x_{f_2}^{l-1}) - r(x_{f_2}^{l-1}, x_{f_2}^l)] < 0 \quad (20)
\]

From which, under the assumption \( \gamma_{b_i} = \gamma \) for all \( b \), we have:

\[
\sum_{1 \leq l \leq k} r(x_{f_2}^{l-1}, x_{f_2}^l) > \sum_{1 \leq l \leq k} r(x_{f_2}^l, x_{f_2}^{l-1}) \quad (21)
\]

where the term on the LHS is the total resistance of path \( \mathcal{P} \) while the term on the RHS is the total resistance of the reverse path \( \bar{\mathcal{P}} = (x_{f_2}^1, x_{f_2}^2, \ldots, x_{f_2}^B) \) if obtained from \( T \) by reversing all the edges of \( \mathcal{P} \).
i.e., $\mathcal{T}' = \mathcal{T} - \mathcal{P} + \mathcal{P}$, we obtain that $R(\mathcal{T}') < R(\mathcal{T})$ contradicting the hypothesis of minimality of $R(\mathcal{T})$. A sketch of this construction is in Fig. 1b.

In the most general case, i.e., when $\gamma_b$ are different, a set of qLRU-$\Delta$ caches all with the same parameter $q$ is not anymore guaranteed to be locally optimal (previous proof fails because from (20) we cannot deduce (21)). However we can still define a provable locally optimal scheme, if we allow the adoption of different parameters $q$ at different cells for the implementation of the local qLRU policy. In particular by selecting $q_b = (q_0)_{\gamma_b}$ we can force characteristic times $T_{C}^{(b)}$ to be asymptotically equal at different cells. In this case, the direct resistances are $r(x_f,y_f) = \gamma_b$, $r(y_f,x_f) = G_f^{(b)}(x_f)$ when $y$ is chosen to be a parent of $x$. As a consequence we have:

$$\phi(x_f) - \phi(y_f) = r(x_f,y_f) - r(x_f,y_f)$$

and by repeating exactly the same arguments as for the special case $\gamma_b = \gamma$, we can prove (17).

V. CASE STUDIES

As we discussed, qLRU-$\Delta$ can be made to optimize different utility functions $G_f$. In this section we illustrate two specific case studies: hit rate maximization, and delay minimization with CoMP techniques. We first describe what form the general qLRU-$\Delta$ assumes in these cases and then illustrate with some experiments the convergence result in Proposition [IV.1]

A. Hit rate maximization

The gain is simply 1 from a hit and 0 from a miss, i.e.,

$$g_f(x_f,u) = \mathbb{1}(J_{u,f} \neq \emptyset),$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. According to (7) with $\beta = 1$, each BS $b$ with a local copy $(b \in J_{u,f})$ moves the content to the front of the cache with probability

$$p_f^{(b)}(u) = \Delta g_f^{(b)}(x_f(t),u)$$

$$= \mathbb{1}(J_{u,f} \neq \emptyset) - \mathbb{1}(J_{u,f} \setminus \{b\} \neq \emptyset)$$

$$= 1 - \mathbb{1}(J_{u,f} \setminus \{b\} \neq \emptyset)$$

$$= \mathbb{1}(J_{u,f} \setminus \{b\} = \emptyset) = \mathbb{1}(J_{u,f} = \{b\}).$$

The policy then works as follows. Upon a miss ($J_{u,f} = \emptyset$), at least one cache decides to retrieve the content with probability $q$. Upon a hit ($J_{u,f} \neq \emptyset$), the cache serving the content brings it to the front if and only if no other cache could have served it (i.e., $|J_{u,f}| = 1$).

This policy is a slight extension of qLRU-LAZY proposed in [13]. The only minor difference is that under qLRU-LAZY only one cache retrieves the contents upon a miss. qLRU-$\Delta$ allows for some additional flexibility. In what follows, we consider that each cache decides independently to retrieve the copy (and then multiple copies of the same content can be retrieved).

B. Delay minimization with CoMP

Let $h_{b,u}$ denote the signal-to-noise ratio (SNR) of the wireless channel between BS $b$ and user $u$. We assume for simplicity that $\{h_{b,u}, b \in I_u\}$ are i.i.d. random variables with expected value $h$, and we consider $h_{b,u} = 0$, when $u$ is not reachable by the BS $b$ ($b \notin I_u$). We consider BSs can employ a coordinated transmission technique. In particular, BSs in $J_{u,f}$ can cooperate to transmit the file to $u$, and we assume they are able to achieve the aggregate channel capacity
maximum delay on the transmission channel. Similarly, e.g. equal to the sum of the backhaul delay and the problem (6), we can simply consider frame this goal according to our reference maximization over the region (see Sect. II).

when a potentially unbounded set of users is distributed when the set of users is finite, and

\[ \sum_{b \in J_{u,f}} h_{b,u} X_f^{(b)}(t) \]

The total expected delay per request is then

\[ D_f(X_f(t)) = \sum_{u=1}^{U} \lambda_{f,u} \mathbb{E}[d_f(X_f(t), u)] , \]

when the set of users is finite, and

\[ D_f(X_f(t)) = \int_{R} \lambda_f(r) \mathbb{E}[d_f(X_f(t), r)] \mu(r) dr, \]

when a potentially unbounded set of users is distributed over the region (see Sect. II).

We want to minimize the delay \( D_f(x_f) \). In order to frame this goal according to our reference maximization problem (6), we can simply consider \( G_f(x_f) \triangleq d_{\text{max}} - D_f(x_f) \), where \( d_{\text{max}} \) is a bound on the retrieval time, e.g. equal to the sum of the backhaul delay and the maximum delay on the transmission channel. Similarly, we consider \( g_f(x_f, u) \triangleq d_{\text{max}} - d_f(x_f, u) \). Note that

\[ \Delta g_f^{(b)}(x_f, u) = (d_{\text{max}} - d_f^{(b)}(x_f, u)) \]

\[ = (d_{\text{max}} - d_f^{(b)}(x_f \oplus e^{(b)}, u)) \]

\[ = d_f^{(b)}(x_f \oplus e^{(b)}, u) - d_f^{(b)}(x_f, u) \]

\[ = \begin{cases} 
  d_B + \frac{M}{c(h_{b,u})} - \frac{M}{c(h_{b,u})}, & \text{if } J_{u,f} = \{b\} \\
  \frac{M}{c(h_{b,u})} - \frac{M}{c(h_{b,u})} - c \left( \frac{\sum_{b \neq b'} h_{b,u} X_f^{(b')}(t)}{c(h_{b,u})} \right), & \text{o/w.} 
\end{cases} \]

Note that \( d_{\text{max}} \) cancels out and then the choice of its value is irrelevant for the algorithm.

Remember from our discussion at the end of Sect. III that it is possible to replace \( \Delta g_f^{(b)} \) in (7) with any other function with the same expected value. Given that \( h_{b,u} \) and \( h_{b',u} \) are identically distributed, we can have each BS \( b \) with a local copy \( (b \in J_{u,f}) \) move the content to the front of the cache with probability

\[ p_f^{(b)}(u) = \begin{cases} 
  \beta d_B, & \text{if } J_{u,f} = \{b\} \\
  \frac{\beta M}{c(\sum_{b \neq b'} h_{b',u} X_f^{(b')}(t))} - \frac{\beta M}{c(\sum_{b \neq b'} h_{b',u} X_f^{(b')}(t))}, & \text{o/w.} 
\end{cases} \]

Similarly, from (8), at least one of the BSs without the content (i.e., those in \( J_{u,f} \setminus J_{u,f} \)) decides if storing an additional copy of \( f \) with probability

\[ q_f^{(b)}(u) = \begin{cases} 
  q \delta d_B, & \text{if } J_{u,f} = \emptyset \\
  \frac{q \delta M}{c(h_{b,u} + \sum_{b'} h_{b',u} X_f^{(b')})} - \frac{q \delta M}{c(h_{b,u} + \sum_{b'} h_{b',u} X_f^{(b')})}, & \text{o/w.} 
\end{cases} \]

As above, we consider that each cache decides independently to retrieve an additional copy.

### C. Numerical Results

We consider a topology where \( B = 10 \) base stations are located according to the positions of T-mobile base stations in Berlin extracted from [25]. The BS locations are indicated in Fig. 2. We assume their transmission range is 150m, and spatial user density to be homogeneous, so that each user on average is covered by 5.9 BSs. SNRs have constant values \( h_{b,u} \) and \( h_{b',u} \) identically distributed, we can have each BS \( b \) with a local copy \( (b \in J_{u,f}) \) move the content to the front of the cache with probability

\[ g_f^{(b)}(u) = \begin{cases} 
  q \delta d_B, & \text{if } J_{u,f} = \emptyset \\
  \frac{q \delta M}{c(h_{b,u} + \sum_{b'} h_{b',u} X_f^{(b')})} - \frac{q \delta M}{c(h_{b,u} + \sum_{b'} h_{b',u} X_f^{(b')})}, & \text{o/w.} 
\end{cases} \]

As above, we consider that each cache decides independently to retrieve an additional copy.

![Fig. 2. T-Mobile BS configuration in Berlin.](image)
the delay rate. In the figures we refer to the two cases as qLRU-Δh and qLRU-Δd. For qLRU-Δd, we set β and δ equal to the minimum value that guarantees respectively \( p_f^{(b)}(u) \leq 1 \) and \( q_f^{(b)}(u) \leq q \) for any possible state of the cache \( X_f \). We would like to compare their performance with the corresponding optimal static allocations. Unfortunately, both corresponding optimization problems are NP-hard, but the greedy algorithm has a guaranteed 1/2-approximation ratio for hit ratio maximization \[3\] and for delay minimization \[13\]. We then consider the corresponding static allocations as baselines and denote them respectively as GREEDY-h and GREEDY-d. Note that the greedy algorithm requires complete knowledge of the network and of content popularities, while qLRU-Δ has no such information.

In all our experiments, policies’ simulations have a warm up phase and a measurement phase each consisting of \( 10^8 \) requests.

Figure 3 (right) shows how such distance decreases as \( q \) decreases, indicating that the files GREEDY-h stores tend to be cached longer and longer under qLRU-Δh.

Figure 4 shows the corresponding results for GREEDY-d and qLRU-Δd. The conclusion is the same: as \( q \) decreases qLRU-Δ improves the metric of interest (the delay in this case) achieving performance comparable to those of the optimal static greedy allocation.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we have introduced qLRU-Δ, a general-purpose caching policy that can be tuned to optimize different performance metrics in a dense cellular network. Recently \[27\], we discovered that the same approach can be applied to a different application scenario, i.e., similarity caching systems, in which a user request for an object \( o \) that is not in the cache can be (partially) satisfied by a similar stored object \( o' \), at the cost of a loss of user utility. This cost can be expressed as function of the set of objects currently stored in the cache, similarly to how in this paper the gain is a function of the set of BSs storing the content.

Under stationary request processes, the smaller \( q \) is, the better qLRU-Δ performs. When content popularities and/or user densities vary over time, the caching policy may react too slowly to changes if \( q \) is small. A detailed experimental evaluation in \[13\] using real traces from Akamai suggests that the sweet-spot is for \( q \) values between 0.01 and 0.1, that achieve a good tradeoff between convergence speed and performance. A practical alternative to make the policy more reactive is to use a virtual cache. The virtual cache only stores content ids and it is managed independently from the physical cache, e.g., through a LRU policy. Upon a miss at a physical cache, the content is stored there if and only if its id is present in the virtual cache. Upon hits, the policy updates the state of the cache exactly as qLRU-Δ. Under stationary request traffic, a miss for content \( i \) leads to an insertion with probability \( 1 - e^{-\lambda_i T_{c,v}} \), where \( T_{c,v} \) is the characteristic time of the virtual cache. The virtual cache can be seen as an alternative way to implement a probabilistic insertion (at the cost of introducing a popularity-bias), achieving small insertion probabilities when the virtual cache (and then \( T_{c,v} \)) is small. At the same time, two close requests for a content cause it to be placed immediately in the physical cache, while qLRU-Δ would store it on average after 1/q requests.

Precisely, the greedy static allocation achieves at least 1/2 of the delay savings achievable by the best possible static allocation.

6 The cosine distance between vectors \( u \) and \( v \) is given by \( dist(u,v) = 1 - \frac{\langle u,v \rangle}{\|u\|\|v\|} \), where \( \langle \cdot, \cdot \rangle \) denotes the inner product.
This variant reacts faster and it is then more suited for non-stationary settings.

$q\text{LRU-}\Delta$ responds to hits in a binary way: the content is moved to the front or maintained in the same position. The dynamic performance of the policy may probably be improved by introducing a list-based variant [28], where the cache is organized in a number of ranked lists and a content is promoted to a higher-priority list upon a hit. The marginal gain of the copy can affect the probability of the content to be randomly promoted to the next list, or the number of lists the content advances by.

Another interesting research direction is to extend $q\text{LRU-}\Delta$ to operate with heterogeneous content sizes. This can be probably achieved by making the update probability inversely proportional to the content size, similarly to what done in [29].

This work was partly funded by the French Government (National Research Agency, ANR) through the “Investments for the Future” Program reference #ANR-11-LABX-0031-01.

**REFERENCES**

[1] “Cisco visual networking index: Global mobile data traffic forecast update, 2017?!2022 white paper,” CISCO, Tech. Rep., February 2019.

[2] 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, Feb. 2012.

[3] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, “Femtocaching: Wireless video content delivery through distributed caching helpers,” IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8402–8413, 2013.

[4] M. Garetto, E. Leonardi, and V. Martina, “A unified approach to the performance analysis of caching systems,” ACM Trans. Model. Perform. Eval. Comput. Syst., vol. 1, no. 3, pp. 12:1–12:28, May 2016.

[5] C. Cox, An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications. John Wiley & Sons, 2012.

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

[7] 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, Oct 2014.

[8] K. Naveen, L. Massoulie, E. Baccelli, A. Carneiro Viana, and D. Towsley, “On the interaction between content caching and request assignment in cellular networks,” in 5th Workshop on All Things Cellular: Oper. Appl., and Challenges. ACM, 2015.

[9] A. Chattopadhyay and B. Blaszczyszyn, “Gibbsian on-line distributed content caching strategy for cellular networks,” IEEE Trans. on Wireless Communications, vol. 17, no. 2, 2018.

[10] B. Blaszczyszyn and A. Gionisvidis, “Optimal geographic caching in cellular networks,” in IEEE ICC 2015, June 2015, pp. 3358–3363.

[11] K. Avrachenkov, J. Goseling, and B. Serbetci, “A low-complexity approach to distributed cooperative caching with geographic constraints,” Proc. ACM Meas. Anal. Comput. Syst., vol. 1, no. 1, pp. 27:1–27:25, Jun. 2017.

[12] W. C. Ao and K. Psounis, “Distributed caching and small cell cooperation for fast content delivery,” in MobiHoc. ACM, 2015, pp. 127–136.

[13] A. Tuholokova, G. Neglia, and T. Spyropoulos, “Optimal cache allocation for femto helpers with joint transmission capabilities,” in 2017 IEEE International Conference on Communications (ICC), May 2017, pp. 1–7.

[14] Z. Shen, J. Lee, T. Q. S. Quek, and M. Kountouris, “Cooperative caching and transmission design in cluster-centric small cell networks,” IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 3401–3415, May 2017.

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

[16] A. Giovanidis and A. Avranas, “Spatial multi-lru caching for wireless networks with coverage overlaps,” SIGMETRICS Perform. Eval. Rev., vol. 44, no. 1, pp. 403–405, Jun. 2016.

[17] G. S. Paschos, A. Destounis, L. Vigneri, and G. Iosifidis, “Learning to cache with no regrets,” in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, April 2019, pp. 235–243.

[18] E. Leonardi and G. Neglia, “Implicit coordination of caches in small cell networks under unknown popularity profiles,” IEEE Journal on Selected Areas in Communications, vol. 36, no. 6, pp. 1276–1285, June 2018.

[19] H. Che, Y. Tung, and Z. Wang, “Hierarchical Web caching systems: modeling, design and experimental results,” Selected Areas in Communications, IEEE Journal on, vol. 20, no. 7, pp. 1305–1314, Sep 2002.

[20] H. P. Young, “The Evolution of Conventions,” Econometrica, vol. 61, no. 1, pp. 57–84, January 1993.

[21] R. Fagin, “Asymptotic miss ratios over independent references,” Journal of Computer and System Sciences, vol. 14, no. 2, pp. 222 – 250, 1977.

[22] M. Garetto, E. Leonardi, and S. Traverso, “Efficient analysis of caching strategies under dynamic content popularity,” in IEEE INFOCOM 2015, April 2015, pp. 2263–2271.

[23] P. R. Jelenkovic, “Asymptotic approximation of the move-to-front search cost distribution and least-recently used caching fault probabilities,” The Annals of Applied Probability, vol. 9, no. 2, pp. 430–464, 1999.

[24] C. Fricker, P. Robert, and J. Roberts, “A versatile and accurate model. Perform. Eval. Rev., vol. 1, no. 3, pp. 12:1–12:28, May 2016.

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

[26] “Openmobilenetwork.” [Online]. Available: http://map.openmobilenetwork.org/