Spatial multi-LRU: Distributed Caching for Wireless Networks with Coverage Overlaps

Anastasios Giovanidis, Apostolos Avranas

Spatial multi-LRU: Distributed Caching for Wireless Networks with Coverage Overlaps

Anastasios Giovanidis, Apostolos Avranas

***Abstract*—This article introduces a novel family of decentralised caching policies, applicable to wireless networks with finite storage at the edge-nodes (stations). These policies, that are based on the *Least-Recently-Used* replacement principle, are here referred to as spatial *multi-LRU*. They update cache inventories in a way that provides content diversity to users who are covered by, and thus have access to, more than one station. Two variations are proposed, the *multi-LRU-One* and *-All*, which differ in the number of replicas inserted in the involved caches. We analyse their performance under two types of traffic demand, the Independent Reference Model (IRM) and a model that exhibits temporal locality. For IRM, we propose a Che-like approximation to predict the hit probability, which gives very accurate results. Numerical evaluations show that the performance of multi-LRU increases the more the multi-coverage areas increase, and it is close to the performance of centralised policies, when multi-coverage is sufficient. For IRM traffic, multi-LRU-One is preferable to multi-LRU-All, whereas when the traffic exhibits temporal locality the -All variation can perform better. Both variations outperform the simple LRU. When popularity knowledge is not accurate, the new policies can perform better than centralised ones.**

***Index Terms*—Wireless, Cache, LRU, Information Centric Networking, Hit Probability, Popularity, IRM, Temporal locality.**

I. Introduction

THE design of today's and future networks is characterised by a paradigm shift, from a host-centric communication architecture, towards an Information Centric Networking (ICN) one [39]. Following this novel concept, network nodes are equipped with storage capacity where data objects can be temporarily cached and retrieved [32]. In this way information can be made available closer to the user, it can be accessed reliably [36] with minimum delay, and possibly with a quality adaptable to the users' preferences, as envisioned in the case of multimedia files. The principal benefits are the partial elimination of redundant traffic flows at the core network by serving demands from intermediate nodes, as well as reduced latency of service [35]. The edge-nodes constitute a very important part of the ICN architecture, since it is where the wireless users directly have access to. When these nodes are equipped with storage capability, download path length is minimised [15].

In this work, we consider the wireless edge of a content centric network, which consists of a set of transmitting nodes taking fixed positions on a planar area, and a set of users dynamically arriving at this area and asking for service. The set of transmitters can refer to base stations (BSs) of a cellular network, small stations of heterogeneous networks, WIFI hotspots, or any other type of wireless nodes that can provide access to users. A user can be covered by multiple of these nodes, but she/he will be served by only one. Each node is equipped with memory of some given size.

An important question is how to best manage the available edge-memories, in order to maximise the hit probability of user-demands. We define the hit probability as the probability that a user finds her/his demand cached in the memory of one of the covering cells. By managing, we mean to choose a policy that decides which objects to install in each cache and how each cache inventory is updated over time.

Given the possibility for multi-coverage, cache management should target two, somewhat conflicting, goals: On the one hand make popular objects, which attract the large bulk of demands, generously available at many geographic locations. On the other hand, make good use of multi-coverage, by filling the memory caches in a way that provides large object variety to each user, so that also less popular objects can be found in the caches. Additionally, since wireless nodes (BSs) are scattered over a very large area and are of considerable number, related operations should be distributed as in [29], [9], [26], and centralised solutions should be avoided.

A. Related research

Single Cache: There exists a variety of cache placement policies that apply to *single caches*, when no coverage overlap is considered. These include the Least Frequently Used (LFU), the Least Recently Used (LRU), and their variations. Specifically LRU has been extensively studied and approximations to the hit probability have been proposed, like the one from Dan and Towsley [13]. Che et al proposed in 2002 [11] a simple approximation for the (single-)LRU under the Independent Reference Model (IRM) [12], which results in an analytic formula for the hit probability with excellent fit to simulations. This fitness is theoretically explained by Fricker et al in [18]. Application of the Che approximation under more general traffic conditions, to variations of the LRU for single caches

A. Giovanidis (anastasios.giovanidis@lip6.fr) conducted this research while in the CNRS-LTCI laboratory, Télécom ParisTech, 23 avenue d'Italie, 75013, Paris, France; he is now with the CNRS-LIP6 laboratory of the University Pierre et Marie Curie (UPMC), Sorbonne Universities, 4 place Jussieu, Boite courrier 169, Couloir 26-00, Bureau 107, 75252 Paris Cedex 05, France.

A. Avranas (apostolos.avranas@huawei.com) conducted part of this research as student of Aristotle University of Thessaloniki, Greece, and of Télécom ParisTech, Paris, France. He is currently with the Mathematical and Algorithmic Sciences Lab, France Research Center, Huawei Technologies Co. Ltd., Arcs de Seine Bat. A, 20 quai du Point du Jour 92100 Boulogne Billancourt, France.

Preliminary versions of this material have been presented at ACM SIGMETRICS '16 / IFIP Performance [20] and IEEE ISTC '16 [3].

as well as networks of caches, is provided by Martina et al [28]. In that work, and further in Elayoubi and Roberts [15], it is shown that for mobile networks, application of pre-filtering improves the performance of LRU.

Multiple Caches: The problem of optimal content placement, when network areas are covered by more than one station has also been recently studied. A number of pro-active caching policies have been proposed, where the cache inventories are pre-filled by content, based on knowledge of the content popularity distribution and additional network-related information. Golrezaei et al [19] find the optimal content placement that maximises hit probability, when full network information (popularity, node and user positions) is available. They formulate a binary optimisation problem and propose approximation and greedy algorithms for its solution. Using reduced information (content popularity, coverage probability), Błaszczyszyn and Giovanidis [7] provide a randomised strategy that maximises the hit probability. Poularakis et al. [33] formulate and solve the joint content placement and user association problem that maximises the fraction of content served by the caches of the edge-nodes. Araldo et al. [2] propose joint cache sizing/object placement/path selection policies that consider also the cost of content retrieval. Recently, Naveen et al. [29] have formulated the problem in a way to include the bandwidth costs, and have proposed an online algorithm for its solution. Further distributed replication strategies that use different system information are proposed by Borst et al [9], and also by Leconte et al [26]. The problem of optimal request routing and content caching for minimum average content access delay in heterogeneous networks is studied by Dehghan et al in [14].

Traffic: There can be strong dependencies between content demands, objects can have a finite lifespan, and new ones can appear anytime. These phenomena constitute the *temporal locality* (not captured from the standard IRM traffic model). Such type of traffic was studied for (single-)LRU initially by Jelenković and Radovanović [23], and recently using also statistics from user measurements, by Traverso et al [38] and Olmos et al [31].

Point Processes: The cache management problem for cellular networks has also been approached using point process modelling of the network node positions. Bastug et al. [6] find the outage probability and content delivery rate for a given cache placement. Furthermore, Tamoor-il-Hassan et al [37] find the optimal station density to achieve a given hit probability, using uniform replication. The policy in [7] can also be applied for point process BS placement.

B. Contributions

This work has the following contributions to the subject of caching at the network edge.

- *Main contribution:* It introduces (Sec. II) a new family of decentralised caching policies, which exploit multi-coverage, called *spatial multi-LRU*. Specifically, two variations of this family are studied, namely multi-LRU-One and -All. These policies constitute an extension of the classic (single-)LRU, to cases where objects can be retrieved by more than one cache. The work investigates how to best adapt the actions of update, insertion and eviction of content for multiple caches.

- The modelling takes geometry and time explicitly into consideration for the analysis of caching policies. Specifically, it investigates a three-dimensional model (two-dimensional space and time). In this, stations have a certain spatial distribution (modelled by Point Processes) and coverage areas may overlap, allowing for multi-coverage. Furthermore, it is a dynamic model, where users with demands arrive over time at different geographic locations (Sec. III).

- The hit probability performance of the new policies is evaluated for two types of traffic: (a) IRM (Sec. IV), and (b) traffic with temporal locality (Sec. V). Specifically for the case of IRM, we initiate from the Che approximation to derive new analytic solutions (Sec. IV-B). Two additional approximations are made here, namely the Cache Independence Approximation (CIA) for multi-LRU-One, and the Cache Similarity Approximation (CSA) for multi-LRU-All, that allow for simple but accurate analytic formulas.

- The performance of the policies is numerically evaluated. Verification for the Che-like approximations (IRM), as well as further comparison of the multi-LRU policies with other policies from the literature, under both traffic inputs, are provided in Sec. VI. For comparison we consider distributed as well as centralised policies that use various network information.

Important conclusions: For IRM, the multi-LRU-One always performs better than the -All variation. For traffic with temporal locality, the multi-LRU-All can perform better than -One in cases where sufficient memory is available. Both policies outperform the (single-)LRU and perform close to centralised policies for IRM traffic with significant multi-coverage. In the case of temporal locality it is shown that multi-LRU can better adapt to popularity changes compared to policies which depend on popularity estimates and content prefetching.

II. Caching and its Management

Caching policies can profit from various system information related e.g. to user traffic, node positions and coverage areas, or caching decisions of neighbouring nodes. Specifically, based on the available knowledge on content popularity, cache management policies can be grouped into two categories:

(i) *Policies with per-reQuest updates* (POQ). For these, file popularity information is not available. Updates of the cache content are done locally and are triggered by the users on a per-request basis. The Least Frequently Used (LFU), as well as the Least Recently Used (LRU and q-LRU) policies for single cache fall in this category.

How **LRU** works: Given an isolated cache of size K, the policy keeps the K most recently demanded objects. The first position of the cache is called Most Recently Used (MRU) and the last one Least Recently Used (LRU). When a new demand arrives, there are two options. (a. *Update*) The object demanded is already in the cache and the policy updates the object order by moving it to the MRU position. (b. *Insertion*) the object is not in the cache and it is inserted as new in the MRU position, while the object in the LRU position is *evicted*. In this work we call this policy, *single-LRU*.

(ii) *Policies with Popularity updates* (POP), where exact or estimated information over content popularities is available, and is used to infrequently update cache inventories by prefetching. This category covers the Most Popular Content (MPC) caching strategy, as well as policies that result from solutions of optimisation problems with a-priori knowledge of additional system information, e.g. the *Greedy Full Information (GFI)* [19], and the *Probabilistic Block Placement (PBP)* [7]. Due to the extra information, POP are expected to have higher hit-probability than POQ (but this is not always true).

A. Spatial multi-LRU

This work introduces a novel family of distributed cache management POQ policies that profit from multi-coverage. These are the *spatial multi-LRU* policies and are based on the single-LRU. The main idea is that, since a user can check all the caches of covering BSs for the demanded object and download it from any one that has it in its inventory, cache updates and object insertions can be done more efficiently than by just applying single-LRU independently to all caches. The fact that the user triggers a cache's update/insertion action, allows each cache to be indirectly informed about the inventory content of its neighbours. Variations of the multi-LRU family differ in the number of inserted contents in the network, after a missed content demand. Differences can also appear in the update phase.

• **multi-LRU-One:** Action is taken in only *one* cache out of the covering $m \geq 1$. (a. Update) If the content is found in a non-empty subset of the m caches, only one cache from the subset is used for download and, for this, the content can be moved to the MRU position. (b. Insertion) If the object is not found in *any* cache, it is inserted only in one, while its least-recently-used object is evicted. This one cache can be chosen as the closest to the user, a random one, or from some other criterion. (Here, we choose the *closest* node).

• **multi-LRU-All:** Insertion action is taken in *all* m caches. (a. Update) If the content is found in a non-empty subset of the m caches, all caches from this subset are updated. (b. Insertion) If the object is not found in *any* cache, it is inserted in all m. A variation based on q-LRU can be proposed, where the object is inserted in each cache with probability $q > 0$.

The motivation behind the different versions of the multi-LRU policies is the following. When a user has more than one opportunity to be served due to multi-coverage, she/he can benefit from a larger cache memory (the sum of memory sizes from covering nodes.). In this setting, the optimal insertion of new content and update actions are not yet clear. If multi-LRU-One is applied, a single replica of the missed content is left down in one of the $m > 1$ caches, thus favouring diversity among neighbouring caches. If multi-LRU-All is used, m replicas are left down, one in each cache, thus spreading faster the new content over a larger geographic area (the union of m covering cells), at the cost of diversity. q-multi-LRU-All is in-between the two, leaving down a smaller than m number of replicas. A-priori, it is unclear which one will perform better with respect to hit probability.

The performance largely depends on the type of incoming traffic. For fixed object catalogue and stationary traffic, diversity in the cache inventories can be beneficial, whereas for time-dependent traffic with varying catalogue, performance can be improved when many replicas of the same object are available, before its popularity perishes.

III. Network Model

Wireless multi-coverage: For the analysis, the positions of transmitters coincide with the atoms from the realisation of a 2-dimensional *stationary* Point Process (PP), $\Phi_b = \{x_i\}$, indexed by $i \in \mathbb{N}_+ = \{1, 2, \ldots\}$, with intensity $\lambda_b > 0$ in $[m^{-2}]$. In this setting, the type of PP can be general, however we consider here:

- A homogeneous *Poisson PP* (PPP) $\Phi_{b,P}$ with intensity measure $\mathbb{E}\left[\Phi_{b,P}(A)\right] = \lambda_b |A|$, for some area $A \subset \mathbb{R}^2$, where $|A|$ is the surface of A.

- A *square lattice* $\Phi_{b,L} = \eta\mathbb{Z}^2 + u_L$, $\mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$, whose nodes constitute a square grid with edge length $\eta > 0$, randomly translated by a vector u_L that is uniformly distributed in $[0, \eta]^2$ (to make $\Phi_{b,L}$ stationary). Its intensity is equal to $\lambda_b = \eta^{-2}$.

There are two different planar areas (*cells*) associated with each atom (BS) x_i. The first one is the *Voronoi cell* $\mathcal{V}(x_i) \subset \mathbb{R}^2$. Given a PP, the Voronoi tessellation divides the plane into *non-overlapping* planar subsets, each one associated with a single atom. A planar point z belongs to $\mathcal{V}(x_i)$, if atom x_i is the closest atom of the process to z. In other words, $\mathcal{V}(x_i) = \left\{z \in \mathbb{R}^2 : |z - x_i| \leq |z - x_j|\,,\ \forall x_j \in \Phi\right\}$.

The second one is the *coverage cell* $\mathcal{C}_i$. Each transmitter node $x_i \in \Phi_b$ has a possibly random area $\mathcal{C}_i$ of wireless coverage associated with it. When users arrive inside the coverage cell of x_i they can be served by it, by downlink transmission. In general $\mathcal{C}_i$ is different from $\mathcal{V}(x_i)$. Coverage cells can overlap, so that a user at a random location may be covered by multiple BSs, or may not be covered at all. The total coverage area from all BSs with their coverage cells is $\Psi = \bigcup_{i \in \mathbb{N}_+} \{x_i + \mathcal{C}_i\}$ (see [4, Ch.3]).

Due to stationarity of the PP Φ_b, any planar location $y \in \mathbb{R}^2$ can be chosen as reference for the performance evaluation of the wireless model. This reference is called the *typical location* o, and for convenience we use the Cartesian origin $(0, 0)$.

Because of the random realisation of the BS positions and the random choice of the reference location o, the number of BS cells covering o is also random. This *coverage number* random variable (r.v.) $\mathcal{N}$ (see [7], [25]) depends on the PP Φ_b and the downlink transmission scheme, with mass function

$$p_m := \mathbb{P}\left[\mathcal{N} = m\right], \qquad m = 0, 1, \ldots, M, \tag{1}$$

where $M \in \mathbb{N}_+ \cup \{\infty\}$. It holds, $\sum_{m=1}^{M} p_m = 1$.

The choice of the coverage model determines the shape of the coverage cells and consequently the values of the coverage probabilities p_m. In this work the choice of $\mathcal{C}_i$ is left to be general (for the evaluation, specific models are considered). Special cases include: (1) the SINR *Model* and (2) the SNR *or Boolean Model*. Both models consider the coverage cell $\mathcal{C}_i$ of x_i, as the set of planar points for which the received signal quality from x_i exceeds some threshold value T. The motivation is that T is a predefined signal quality, above which

the user gets satisfactory Quality-of-Service. The difference between these two is that the SINR model refers to networks with interference (e.g. when BSs serve on the same OFDMA frequency sub-slot), whereas the SNR model, to networks that are noise-limited (e.g. when neighbouring BSs operate on different bandwidth, by frequency reuse). For the Boolean model the $\mathcal{C}_i$ is a ball $\mathcal{B}(x_i, R_b)$ of fixed radius R_b centred at x_i. It coincides with the SNR model, when no randomness of signal fading over the wireless channel is considered. A more detailed presentation of the different coverage models can be found in [4], [8] and [5].

Storage: We consider the case where a cache memory of size $K \geq 1$ is installed and available on each transmitter node x_i of Φ_b. (All content files are considered of equal size, see below). The memory inventory of node x_i at time t is denoted by $\Xi_i(t)$ and is a (possibly varying over time) subset of the content catalogue $\mathcal{F}(t)$, with number of elements $|\Xi_i(t)| \leq K$.

Traffic Models: Each user arrives at some point in space and time, with a request for a specific data object. The arrivals are assumed spatially independent. We model the users by a *marked space-time* Point Process in $\mathbb{R}^2 \times \mathbb{R} \times \mathbb{N}$, $\Phi_u = \{(\psi_i, t_i, z_i)\}$, where ψ_i takes values on the Euclidean plane, and the time t_i of arrival occurs at some point on the infinite time axis. The mark z_i takes as values the indices of the files/objects $j : c_j \in \mathcal{F}(t)$. Service time is considered fixed and equal to unity but it will not play any role in the analysis. In this work, we evaluate the caching policies under the following two traffic models:

- A spatially homogeneous version of the *Independent Reference Model* (IRM) [16] (Sec. IV).

- Traffic that exhibits *temporal locality*, like the Shot Noise Model (SNM) [27] (Sec. V).

Typical user: The network performance is evaluated at the *typical user* u_o, who - due to stationarity of the PPP - will be representative of any user of the process. We suppose that this user appears at the Cartesian origin $(0, 0)$, at time $t_o = 0$. In this way, the *typical user* coincides with the *typical location* o of the process Φ_b at time $t = 0$.

The model described so far is illustrated in Fig.1(a) for the case of Poisson placement of transmitters $\Phi_{b,P}$ with Poisson arrivals Φ_u, and in Fig.1(b) for the case of a square lattice $\Phi_{b,L}$ with Poisson arrivals Φ_u. We also provide the reader with a list of symbols in Table I.

TABLE I
SYMBOLS

Φ_b	Point Process of transmission nodes $\{x_i\}$
$\Phi_{b,P}$, $\Phi_{b,L}$	Poisson and Lattice position of $\{x_i\}$
Φ_u	Point Process of users marked by object $\{(\psi_i, t_i, z_i)\}$
λ_b	intensity of transmission nodes $[m^{-2}]$
λ_u	intensity of users $[m^{-2}sec^{-1}]$
A	planar area
$\mathcal{V}(x_i)$	Voronoi cell of node x_i
$\mathcal{C}_i$	coverage cell of node x_i
R_b	radius of coverage
p_m	probability of coverage by m nodes
$\mathcal{F}$	object catalogue of size F
a_j	popularity of object $c_j \in \mathcal{F}$
o, u_o	Typical location and typical user
K	size of cache memory
$\Xi_i(t)$	inventory of cache on BS x_i at time t
Φ_c	Point Process of new content arrivals
λ_c	intensity of new content arrivals $[obj\ day^{-1}]$
τ_n	lifespan of content n
v_n	volume (total demands) of content n
g_n	popularity shape of content n

IV. MULTI-LRU UNDER SPATIAL IRM TRAFFIC

The new policies are first evaluated under spatial IRM traffic [16], which has the following properties:

(i) The Φ_u is a homogeneous Poisson Point Process (PPP) in both space and time, with intensity $\lambda_u > 0$ in $[m^{-2}sec^{-1}]$. Given a planar area A, the arrival rate of users in this area, with any request, is equal to $\lambda_u|A|$ in $[sec^{-1}]$. All users within the area take their positions independently and uniformly.

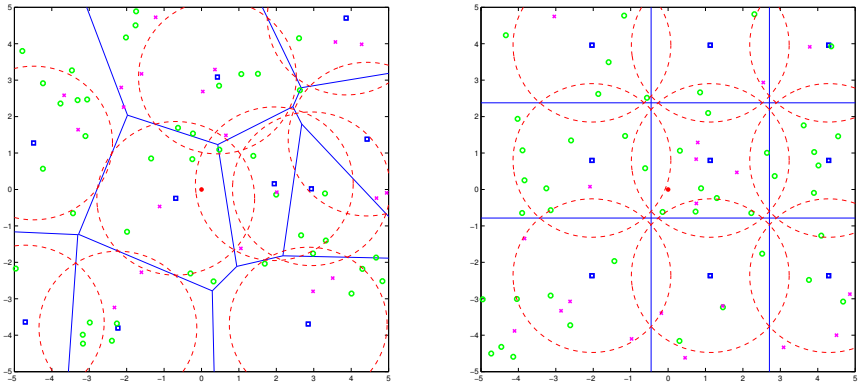
(ii) The catalogue of available files/objects is fixed over time $\mathcal{F}(t) := \mathcal{F}$, and has finite size F. The elements of $\mathcal{F}$ are $\{c_1, \ldots, c_F\}$. We additionally consider that all objects have the same size, normalised to 1. Cases of unequal size will not be treated in this work, but we can always assume that each file can be divided into chunks of equal size, so the same analysis can still be applied.

(iii) The probability a_j that a user requests object $c_j \in \mathcal{F}$ (i.e. the object *popularity*) is constant over time, (can be) known, and independent of all past requests. Objects in $\mathcal{F}$ are ordered by popularity: c_1 is the most popular, c_2 the second most popular and so on. The popularity of c_j is $a_j > 0$, and to be consistent with the ordering, we also have $a_1 \geq a_2 \geq \ldots \geq a_F$. For every popularity distribution it obviously holds, $\sum_{j=1}^{F} a_j = 1$. Then the marks z_i are i.i.d. random variables distributed as Z with mass function $\{a_j\}$. A consequence is that the users that request object $c_j \in \mathcal{F}$ form a homogeneous space-time PPP with intensity $a_j\lambda_u$ $[m^{-2}sec^{-1}]$ (independent thinning of Φ_u).

Without loss of generality, we will consider (especially in the simulations) that the distribution has a Zipf probability mass function, although the analysis holds for general $\{a_j\}$. This is motivated by traffic measurements showing that data-object popularity in the WWW follows a power law [22], [30]. In such case, the probability that a user asks for c_j is equal to $a_j = D^{-1}j^{-\gamma}$, $j = 1, \ldots, F$. Here, γ is the Zipf exponent, often satisfying $\gamma < 1$, so that $a_1/a_2 = 2^{\gamma} < 2$. The normalisation factor is equal to $D := \sum_{j=1}^{F} j^{-\gamma}$.

Hit performance upper bound: As mentioned already, the performance measure of the caching policies is the hit probability. We can already provide an upper bound for any POP policy under IRM traffic. The bound requires knowledge over the content popularity and coverage number, like the PBP [7]. The main idea is that the hit probability of a user covered by m cells is maximised if these m inventories have distinct entries, so that the user has the maximum choice. Hence if the mK most popular objects from the set $\mathcal{F}$ are installed,

$$P_{hit} \leq \sum_{m=1}^{M} p_m \sum_{j=1}^{F} a_j \mathbf{1}_{\{1 \leq j \leq mK\}} = \sum_{m=1}^{M} p_m \sum_{j=1}^{mK} a_j. \qquad (2)$$



(a) Poisson Transmitters/Boolean Coverage (b) Lattice Transmitters/Boolean Coverage

Fig. 1. A realisation of the introduced model for $t = 0$ and a window of size 10×10 $[m^2]$. In both subfigures, user arrivals are modelled by a PPP with $\lambda_u = 0.6$ $[m^{-2}sec^{-1}]$. The users choose between two objects that have popularities $a_1 = 0.65$ (users with "o"), $a_2 = 0.35$ (users with "x"). The typical user is shown at the Cartesian origin $(0, 0)$ (thicker "o"). (a) The transmitters (squares) are modelled by a PPP with $\lambda_b = 0.1$ $[m^{-2}]$. (b) The transmitters (squares) are modelled by a Square Lattice PP with $\eta = \lambda_b^{-1/2} = 1/\sqrt{0.1}$ $[m]$. In both figures, we assume the Boolean model for coverage, with $R_b = 2\eta/3$ $[m]$. In this realisation, the typical user is covered by two cells in the PPP case and by a single one in the Lattice case.

A. Che approximation for IRM traffic: Single Cache

The mathematical analysis of LRU policies is complicated, due to the different inter-arrival times for different content and the update/insertion policy. However, Che et al provided in 2002 [11] an analysis and a simple approximation for the single-LRU cache, which results in an analytic formula for the hit probability P_{hit} with excellent fit to simulations. In the following, we explain in short the idea and, after, apply it to the multi-LRU policies.

The approximation is based on the so-called *characteristic time* T_C. Given a cache of size K under single-LRU replacement, if at time $t = 0$ an arrival of object c_j occurs, then this will be positioned at the MRU place, either due to a. Update, or due to b. Insertion. This object is removed from the cache if at least K different objects arrive, before a new demand for object c_j at time $s_j > 0$. The reason is that, each arrival of a new object moves c_j one position away from the MRU and closer to the LRU. Che et al approximate the eviction time of an object by a deterministic quantity, equal for all objects to the characteristic time T_C. This is found by solving

$$\sum_{i=1}^{F} \mathbb{P}(s_i < T_C) = K \qquad (Che\ approximation), \quad (3)$$

using a fixed point procedure, where s_i is the first arrival time of object c_i, $i \neq j$, after $t = 0$. The summation in (3) is taken over the entire $\mathcal{F}$, which is also part of the approximation. It works well for a large number F of objects, each one of which having a small portion of the popularity. For IRM traffic, the inter-arrival times are exponentially distributed, hence for an area A covered by a single cache, $\mathbb{P}(s_i < T_C) = 1 - e^{-\lambda_u |A| a_i T_C}$. The time-average probability that an object c_j is in the cache is $\mathbb{P}(c_j \in \Xi) = \mathbb{P}(s_j < T_C)$, hence

$$\mathbb{P}(c_j \in \Xi) \overset{IRM}{=} 1 - e^{-\lambda_u |A| a_j T_C} \overset{PASTA}{=} P_{hit}(j). \quad (4)$$

The fact that, for IRM traffic $\mathbb{P}(c_j \in \Xi) = P_{hit}(j)$, is due to the PASTA property of Poisson arrivals. Finally, the approximation for the total hit probability is,

$$P_{hit} = \sum_{j=1}^{F} a_j P_{hit}(j). \quad (5)$$

B. Che-like approximation for multi-LRU

We will use the approach of Che for the single-LRU, to derive here similar approximations of the multi-LRU cache management policies, for the network model described in the previous section. (To provide more intuition on this general approach, a similar analysis for a network with only two caches is given in the Appendix).

Consider an arrival of user u_o at the Cartesian origin $\psi_o = (0, 0)$ at time $t_o = 0$, who requests for object c_j. This is the *typical user*, who is covered by a number $m_o \geq 0$ of BSs, a realisation of the r.v. $\mathcal{N}_o$ with mass function $\{p_m\}$. A common characteristic time T_C is assumed for all caches of the network, due to stationarity of all processes. We focus on the cache of a specific x_i among the m_o covering BSs, for which definitely $o \in \mathcal{C}_i$. The probability that user u_o finds the requested content in the cache of x_i, is calculated using the following arguments: There is a previous user u_{-1} requesting for the same object c_j, who arrived in an area $\mathcal{S}_{-1}$ (that varies depending on the type of multi-LRU policy). The u_{-1} is covered by x_i definitely (otherwise the user will not influence Ξ_i) and possibly some other stations, the total number of which is $\tilde{m}$ (the realisation of another r.v. $\mathcal{N}_{-1}$). Since we know that u_{-1} is at least covered by one station (the x_i), the distribution of $\mathcal{N}_{-1}$ has mass function

$$\tilde{p}_{\tilde{m}} = p_{\tilde{m}}(1 - p_0)^{-1}, \qquad \tilde{m} = 1, \ldots, M. \quad (6)$$

Suppose this user arrived at $t_{-1}^- \in [t_o - T_C, t_o]$, i.e. within the characteristic time (t^- is the time right before t). Then the object is found in $\Xi_i(t_o^-)$ at t_o^-, if (i) either the object was in $\Xi_i(t_{-1}^-)$ and an update was triggered by u_{-1}, or (ii) the object was not cached in any of the $\tilde{m}$ stations and an insertion in

the inventory Ξ_i was triggered. If $m_o > 0$ (otherwise, the user is not covered), we write for $i \in \{1, \dots, m_o\}$

$$P_{hit,i}(u_o) = \mathbb{P}\left(u_{-1} \in (\mathcal{S}_{-1}, |t_o - t_{-1}| < T_C, j)\right) \cdot \cdot \left[\mathbb{P}(c_j \in \Xi_i(t_{-1}^-)) + \mathbb{P}(\bigcap_{\ell=1}^{\tilde{m}} \left\{c_j \notin \Xi_\ell(t_{-1}^-)\right\})\right].$$

For IRM traffic with PASTA, $\mathbb{P}(c_j \in \Xi_i(t_{-1}^-)) = P_{hit,i}(u_{-1})$, and is also independent of the time t and user position ψ, hence we can simply write $P_{hit,i}(j)$. Substitution in the above equation gives,

$$P_{hit,i}(j) = \mathbb{P}\left(u_{-1} \in (\mathcal{S}_{-1}, |t_o - t_{-1}| < T_C, j)\right) \cdot \cdot \left[P_{hit,i}(j) + \sum_{\tilde{m}=1}^{M} \frac{p_{\tilde{m}}}{1 - p_o} \mathbb{P}(\bigcap_{\ell=1}^{\tilde{m}} \left\{c_j \notin \Xi_\ell\right\})\right]. \quad (7)$$

Solving the above over $P_{hit,i}(j)$ provides an expression for the hit probability of object c_j at the cache of node x_i. To find the characteristic time T_C we solve the equation,

$$\sum_{j=1}^{F} P_{hit,i}(j) = K, \qquad i \in \{1, \dots, m_o\}. \quad (8)$$

Finally, the total hit probability is equal to,

$$P_{hit} = \sum_{j=1}^{F} a_j \sum_{m_o=0}^{M} p_{m_o} \left(1 - \mathbb{P}(\bigcap_{\ell=1}^{m_o} \left\{c_j \notin \Xi_\ell\right\})\right). \quad (9)$$

We note that $\mathbb{P}(\bigcap_{\ell=1}^{0} \left\{c_j \notin \Xi_\ell\right\}) = 1$, for $m_o = 0$, in which case, the user surely misses the content.

The main difficulty when dealing with the general case, is that the hit probability of one cache depends on the hit probability of its neighbours and the neighbours of its neighbours. This is because the coverage area of each node has many sub-areas of multi-coverage by different BS subsets, which makes analysis neither easy, nor exact.

• **multi-LRU-One (Che with CIA)** Only the users falling in the Voronoi cell of a node can trigger an action of a. Update or b. Insertion at the cache of that node as long as they are covered. (Here we analyse this version for the Update phase, but other policy variations are possible without significant performance change). Then $\mathcal{S}_o = \mathcal{S}_{-1} = \mathcal{V}(x_i)$ in (7). The coverage cell can be smaller than the Voronoi cell, in which case only the users falling in the intersection of the two trigger cache actions. To avoid dealing with these special cases, we consider coverage cells which fully cover the related Voronoi cells, that is $|\mathcal{C}_i| > |\mathcal{V}_i|$, $\forall i$.

There are the unknown probabilities $\mathbb{P}(\bigcap_{\ell=1}^{\tilde{m}} \left\{c_j \notin \Xi_\ell\right\})$ and $\mathbb{P}(\bigcap_{\ell=1}^{m_o} \left\{c_j \notin \Xi_\ell\right\})$ that need to be calculated. Instead of directly trying to find a solution, we use a *Cache Independence Approximation* (CIA). Based on this, each cache performs single-LRU for the users that arrive within its Voronoi cell. The idea is that, since only the users in the Voronoi cell change the inventory of the related cache, the influence of the neighbouring stations' traffic on the inventory of x_i should be small. Then in (7) we forget the rest $\tilde{m} - 1$ nodes and we replace

$$\mathbb{P}(\bigcap_{\ell=1}^{\tilde{m}} \left\{c_j \notin \Xi_\ell\right\}) \approx \mathbb{P}(c_j \notin \Xi_i), \quad (CIA_1). \quad (10)$$

Furthermore, the independence due to the CIA, has the result that, when the user is covered by m_o stations, her/his hit probability is simply the product of hit probabilities of all these stations. The fact that the Voronoi cells of different stations do not overlap is further in favour of the approximation. Then, in (9)

$$\mathbb{P}(\bigcap_{\ell=1}^{m_o} \left\{c_j \notin \Xi_\ell\right\}) \approx (\mathbb{P}(c_j \notin \Xi_i))^{m_o}, \quad (CIA_2). \quad (11)$$

From the above, the hit probability of each object in Ξ_i is,

$$P_{hit,i}(j) = \mathbb{P}\left(u_{-1} \in (\mathcal{S}_{-1} \in \mathcal{V}(x_i), |t_o - t_{-1}| < T_C, j)\right) \cdot \cdot \left[P_{hit,i}(j) + \mathbb{P}(c_j \notin \Xi_i)\right] \overset{IRM}{=} 1 - e^{-a_j \lambda_u |\mathcal{V}| T_C}, \; i \in \{1, \dots, m_o\}. \quad (12)$$

We used the fact that for IRM $P_{hit,i}(j) = 1 - \mathbb{P}(c_j \notin \Xi_i)$. The characteristic time is found by solving the equation

$$\sum_{j=1}^{F} (1 - e^{-a_j \lambda_u |\mathcal{V}| T_C}) = K. \quad (13)$$

The total hit probability, based on CIA, is,

$$P_{hit} = \sum_{j=1}^{F} a_j \sum_{m_o=0}^{M} p_{m_o} \left(1 - \mathbb{P}(c_j \notin \Xi_i)^{m_o}\right) \overset{(12)}{=} \sum_{j=1}^{F} a_j \sum_{m_o=0}^{M} p_{m_o} \left(1 - e^{-a_j \lambda_u m_o |\mathcal{V}| T_C}\right). \quad (14)$$

Special case: For the PPP model of node positions, the Voronoi cell size is a random variable. We can use for simplicity of the expression the average size of a Voronoi cell, equal to $|\mathcal{V}| = \lambda_b^{-1}$, [4]. In the Boolean coverage model, $|\mathcal{C}| = \pi R_b^2$.

• **multi-LRU-All (Che with CSA)** In this case, users falling on any point inside the coverage cell of x_i can trigger an action of update and insertion at its cache inventory Ξ_i. This means that $\mathcal{S}_o = \mathcal{S}_{-1} = \mathcal{C}_i$, for the hit probability expression in (7).

Again, the unknown probabilities $\mathbb{P}(\bigcap_{\ell=1}^{\tilde{m}} \left\{c_j \notin \Xi_\ell\right\})$ and $\mathbb{P}(\bigcap_{\ell=1}^{m_o} \left\{c_j \notin \Xi_\ell\right\})$ need to be calculated. In this case, we use a different approximation, the *Cache Similarity Approximation* (CSA), which states that inventories of neighbouring caches have the same content. This is motivated by the fact that new content is simultaneously installed in all caches of nodes covering a user, when the user triggers insertion. The approximation is better, the larger the cache size K, because for large memories it takes more time for an object to be evicted after its insertion and similar content stays in all inventories. Then in (7),

$$\mathbb{P}(\bigcap_{\ell=1}^{\tilde{m}} \left\{c_j \notin \Xi_\ell\right\}) \approx \mathbb{P}(\left\{c_j \notin \Xi_i\right\}), \quad (CSA_1). \quad (15)$$

Interestingly, CSA_1 and CIA_1 give the same expression. However, in multi-LRU-All, we do not assume independence, but rather similarity. Then, since neighbouring caches have the same content, the total miss probability when a set of m_o stations cover user u_o is equal to the probability that no user with the same demand arrives within the total area of coverage during the characteristic time T_C (otherwise the content is definitely in all caches, either because of a. Update or b. Insertion. Then, for IRM traffic,

$$\mathbb{P}(\bigcap_{\ell=1}^{m_o}\{c_j \notin \Xi_\ell\}) \approx e^{-a_j\lambda_u|\mathcal{A}_{m_o}|T_C}, \quad (CSA_2). \tag{16}$$

In the above, the total area of coverage from the m_o stations is denoted by $\mathcal{A}_{m_o}$ and its surface is equal to,

$$|\mathcal{A}_{m_o}| \quad = \quad \left|\bigcup_{\ell=1}^{m_o}\mathcal{C}_i\right|, \quad m_o = 0,\ldots,M. \tag{17}$$

It holds $|\mathcal{A}_0| = 0$, for $m_o = 0$. For the Boolean model $|\mathcal{A}_1| = |\mathcal{C}_1| = \pi R_b^2$, while the surface of $\mathcal{A}_{m_o}$ is a superposition of m_o overlapping discs with equal radius R_b.

The hit probability of each object in Ξ_i is found by using CSA in (7), and we get

$$\begin{aligned} P_{hit,i}(j) \quad &= \quad \mathbb{P}\left(u_{-1} \in (\mathcal{S}_{-1} \in \mathcal{C}_i, |t_o - t_{-1}| < T_C, j)\right) \cdot \\ &\quad \cdot [P_{hit,i}(j) + \mathbb{P}(\{c_j \notin \Xi_i\})] \\ &\overset{IRM}{=} \quad 1 - e^{-a_j\lambda_u|\mathcal{C}|T_C}. \end{aligned} \tag{18}$$

We used the fact that for IRM $P_{hit,i}(j) = 1 - \mathbb{P}(c_j \notin \Xi_i)$. For the characteristic time, we solve the equation

$$\sum_{j=1}^{F}(1 - e^{-a_j\lambda_u|\mathcal{C}|T_C}) \quad = \quad K. \tag{19}$$

The total hit probability, based on CSA, is

$$P_{hit} \quad \overset{(16)}{=} \quad \sum_{j=1}^{F} a_j \sum_{m_o=0}^{M} p_{m_o}\left(1 - e^{-a_j\lambda_u|\mathcal{A}_{m_o}|T_C}\right). \tag{20}$$

The difficulty in calculating the approximate hit probability for multi-LRU-All with the above formulas, is to obtain exact values for the total surface $|\mathcal{A}_{m_o}|$. For the PPP special case, this surface is also a random variable, that we can approximate by its mean value for simplicity of the expressions. A method to approximate these quantities is given in V.A. (We refer, again, the reader to the Appendix for the two-cache network example.)

V. Multi-LRU under temporal locality traffic

Although the IRM offers tractability, it is not enough to describe real traffic aspects. In real networks new objects (never requested before) appear, while older ones become obsolete after some time. Furthermore, the popularity of a content does not remain constant but varies over time, and there is dependence between requests of the same object within some time window. All these characteristics are described by the term *temporal locality* [24], [38], [31] (see also [10] for space-locality). Generators of such traffic have been proposed in the literature [1]. A so-called Shot Noise Model (SNM) is presented in [38], [27], which we make use here as a basis of our own traffic model. Under SNM the demand process is a superposition of independent point processes (not necessarily homogeneous), one for each content.

A detailed description of the SNM variation in this work follows. $\mathcal{F}(t)$ is the catalogue (set) of active objects at time t, with cardinality $F(t) := |\mathcal{F}(t)|$. The evolution of the catalogue size is a random process. We assume that the arrival of a new object c_n coincides with the time of its first request t_n. The time instants of these first requests (arrivals) are modelled as a homogeneous PPP Φ_c on $\mathbb{R}$ with intensity $\lambda_c > 0$ $[\frac{objects}{unit-time}]$. Unit-time can be e.g. 1 day.

A pair of r.v.'s is related to each content as an independent mark on the arrival process: (a) The first r.v. denoted by T_n is the n'th content's *lifespan*, which gives the length of time period during which it is requested by users, and after the period's end it becomes obsolete. We could allow for the realisation τ_n to take infinite values but in such option the size of the catalogue would grow indefinitely, unless the popularity of different objects tends to zero fast enough. We let here $\tau_n < \infty$ so that the catalogue size $F(t)$ fluctuates over time and remains finite. The time interval of an object is $\Delta t_n := [t_n, t_n + \tau_n)$. (b) The second r.v. attached to the object c_n is the *volume* V_n (with realisation v_n) i.e. the total number of requests during its life. The pair of values (τ_n, v_n) per object is chosen independently of other objects and in the general case should be drawn from a joint probability distribution with a given density $f_{(T,V)}(\tau, v)$, where T and V are the generic variables. In general the two variables are dependent.

To simplify the traffic model it is assumed here that T and V are independent of each other, i.e. $f_{(T,V)}(\tau, v) = f_T(\tau)f_V(v)$. This simplification has no obvious impact on the performance of the caching policies. Both lifespan and volume follow a Power-law, i.e both T and V are Pareto distributed [30]. The Pareto distribution in both cases has parameter $\beta > 1$ (for the expected value to be finite), and its p.d.f. is given by (here for V) $f_V(v) = \frac{\beta V_{\min}^{\beta}}{v^{\beta+1}}$. Its expected value depends on the values of β and $V_{\min}$ through the expression $\mathbb{E}[V] = \frac{\beta V_{\min}}{\beta-1}$. To guarantee $V \in \mathbb{N}_+$ for the samples, we choose $V_{\min} = 0.5$ and we round to get discrete values. Sampling from a Pareto distribution, generates Zipf-like distributed sizes of objects due to the Power-law behaviour.

Having sampled (τ_n, v_n) for a specific object arriving at t_n, it remains to determine how these v_n requests are positioned within Δt_n. To include additional attributes of temporal locality in the traffic model, we let requests be distributed according to a finite point process (given $V < \infty$) and more specifically a (non-homogeneous) binomial point process (BPP) Ψ_n on $\mathbb{R}^{v_n-1}$ with density function $g_n(t, t_n, \tau_n)$ over t,

$$\Psi_n \sim \text{Binomial}(\Delta t_n, v_n - 1, g_n(t, t_n, \tau_n)). \tag{21}$$

We randomly position only $v_n - 1$ requests, because the first request always coincides with the time of content arrival t_n. The choice of the Binomial distribution further implies that requests take position independently of each other. Since $g_n(t, t_n, \tau_n)$ describes how each of the $v_n - 1$ requests is

distributed within Δt_n according to the function's *shape*, the higher the value of g_n for some t, the more probable it is that a request will appear at that point. The popularity shape is an important aspect of the model. For some c_n it holds

$$g_n(t, t_n, \tau_n) = 0 \quad \text{for} \quad t \notin \Delta t_n, \tag{22}$$

and

$$\int_{t_n}^{t_n+\tau_n} g_n(t, t_n, \tau_n) dt = 1. \tag{23}$$

When the requests follow a homogeneous BPP for some c_n with a given Δt_n, the shape function is *uniform* and takes the expression $g_n(t, t_n, \tau_n) = \tau_n^{-1} \mathbf{1}_{\{t \in \Delta t_n\}}$. For further shape options we refer the reader to [34]. In this reference work, for finite volume per object three shapes are proposed, namely (i) the *logistic*, (ii) the *Gompertz*, and (iii) the *negative exponential*[1]. Applying this to our traffic generator, when a new object arrives it is assigned a shape of index k with probability a_k, the exact value of which is a tuneable parameter.

The spatial (geographic) aspect of traffic plays an important role in influencing the performance of the policies studied here. In our work requests are uniformly positioned on a finite 2D plane. However, the traffic model can be easily extended to incorporate spatial locality.

Based on the above description, characteristic quantities of the generated traffic can be derived.

- *Mean Catalogue Size* $\mathbb{E}[\mathcal{F}(t)]$. Because of the stationarity of the arrival PPP the expected number of active contents (hence catalogue size) does not depend on time $t \in \mathbb{R}_+$,

$$\mathbb{E}[\mathcal{F}(t)] \quad = \quad \lambda_c \mathbb{P}(V > 1) \mathbb{E}[T]. \tag{24}$$

- *Mean Total Number of Requests within* $[0, t]$ $[days]$,

$$N_{req}([0, t]) \quad = \quad t \lambda_c \mathbb{E}[V]. \tag{25}$$

- *Memory-to-Mean-Catalogue-Size-Ratio (MMCSR)* where we omit $\mathbb{P}(V > 1)$, which is just a scaling constant,

$$\rho \quad := \quad \frac{K}{\lambda_c \mathbb{E}[T]}. \tag{26}$$

Hit Upper Bound: Similarly to the IRM traffic, we can derive a (numerical) upper bound for the POP policies under traffic with temporal locality. We consider a scenario where popularities are estimated periodically. Specifically, at time instants $t_n = n \Delta t_{up}$, $n \in \mathbb{Z}$, the caches are updated by some POP policy, using the estimated popularities during the time interval $[t_n - \Delta t_{es}, t_n)$, i.e. Δt_{es} is the window of observation. Let $\mathcal{F}_{mK}(t_n)$ be the set of the mK most requested objects in $[t_n - \Delta t_{es}, t_n)$. Then the upper bound within the time interval $[t_n, t_n + \Delta t_{up})$ is equal to,

$$P_{hit}^{(POP)}[t_n, t_n + \Delta t_{up}) \leq \sum_{m=1}^{\infty} p_m \mathbb{P}(c \in \mathcal{F}_{mK}(t_n)). \tag{27}$$

The more often the algorithm updates the caches the better performance it achieves. But, considering that a cache update will use backhaul and computational resources by the controller, Δt_{up} cannot be too small. We fix $\Delta t_{up} = 1$ day, i.e. the caching policy runs every night when the request load is low [19]. As far as Δt_{es} is concerned there is a "crisp" optimal choice. If it is too big, $\mathcal{F}_{mK}(t_n)$ will possibly include outdated objects. On the other hand, small Δt_{es} can result in excluding even the most popular objects from $\mathcal{F}_{mK}(t_n)$ because they have not been sufficiently requested. The bound in (27) can be evaluated by Monte Carlo simulations. The optimum Δt_{es} $[days]$ can be numerically determined.

[1]By assuming that the lifespan of every content ends when it reaches its $1 - \varepsilon$ of its total views v_i, τ_i can be mapped to the curve parameter λ in [34], which parametrises the speed of change of a content's popularity. In this paper we chose $\varepsilon = 0.02$.

VI. Simulation and Comparison

For the simulations, BSs are placed within a rectangular window of size $A \times B = 12 \times 12$ $[km^2]$. After choosing the BS intensity $\lambda_b = 0.5$ $[km^{-2}]$, their positions are chosen based on the type of network we want to analyse (PPP or Lattice). For PPP, a Poisson number of stations is simulated in each realisation and their positions are set uniformly inside the window. In the case of a Lattice network, the stations are put on a square grid with distance $\eta = 1/\sqrt{\lambda_b} = 1.4142$ $[km]$ from each other. In both types of networks, the average Voronoi size $|\mathcal{V}| = \lambda_b^{-1}$ (see [4]).

We evaluate a Boolean coverage model so that every station covers a disc of radius $R_b \in [0.5, 3]$ $[km]$ with surface $|\mathcal{C}| = \pi R_b^2$. The larger the radius the stronger the multi-coverage effects. The magnitude of coverage overlap can be described by the expected number of BSs covering a planar point, $\overline{N_{BS}} = \mathbb{E}[\text{Number of covering stations}] = \sum_{m=1}^{\infty} m p_m$, where the p_m are the coverage number probabilities for m stations, whose values depend on the node placement and coverage model. For the Boolean PPP case, the probabilities $\{p_m\}$ correspond to a Poisson r.v. with parameter $\nu := \lambda_b \pi R_b^2$ (see [4]). For the Boolean Lattice case, these are found by Monte Carlo simulations. Given the intensity, $\lambda_b = 0.5$, there is a mapping from the Boolean radius R_b to the number $\overline{N_{BS}}$, some values of which are given in Table II.

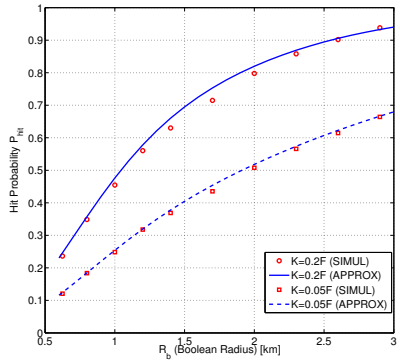
TABLE II
R_b TO $\overline{N_{BS}}$ MAPPING: BOOLEAN PPP AND LATTICE ($\lambda_b = 0.5\, km^{-2}$).

Radius (R_b) $[km]$	PPP ($\overline{N_{BS}}$)	Lattice ($\overline{N_{BS}}$)
0.8	1	1.06
1.13	2	2.12
1.38	3	3.22
1.60	4	4.21
1.78	5	5.32
1.95	6	6.42
2.11	7	7.43
2.26	8	8.44

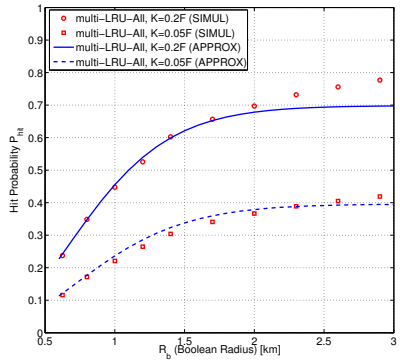
A. IRM traffic

Following the spatial IRM traffic model for the request arrivals, we consider a homogeneous space-time PPP with intensity $\lambda_u = 0.023$ $[m^{-2} sec^{-1}]$, which is approximately equal to 80 $[m^{-2}/hour^{-1}]$ requests - a reasonable value for a busy corner in a city.

The content mark for each request is independently chosen from a catalogue of size $F = 10,000$ objects. The popularities



(a) multi-LRU-One



(b) multi-LRU-All

Fig. 2. Verification of the approximations for the two multi-LRU policies.

of these objects follow a Zipf distribution with parameter $\gamma = 0.78$ (unless otherwise stated). A cache memory of capacity K objects is considered available on each BS. The size K is defined as a proportion of the catalogue size, i.e. $K/F = \rho$, where α is called the *Memory-to-Catalogue-Size Ratio (MCSR)*. (We use different notation than the MMCSR ρ for temporal locality in (26)).

When a user is covered by a station with the requested content in memory, the demand is considered a hit. At the end of the simulation of a large number of realisations for the BS and request point processes (this number chosen over 10,000) the total hit probability is approximated by the frequency of hits (number of hits over number of requests). In the simulations we take considerations over issues related to edge effects that arise from the finite window size.

1) Verification of the approximations: To verify the validity of the proposed approximations, we compare the analytic formulas derived in Section IV-B with the hit probability from simulations, for the Boolean PPP coverage model. For the memory size K, we consider two cases, (a) $\alpha = 0.05$, hence $K = 500$ objects, and (b) $\alpha = 0.2$, hence $K = 2000$ objects.

• **multi-LRU-One**: The total hit probability is evaluated numerically using (14). The characteristic time per cache is found by solving (13) by a fixed point method, where the individual hit probability of each object is given in (12). To guarantee that $|\mathcal{C}| > |\mathcal{V}|$, we need that $\pi R_b^2 > \lambda_b^{-1} \Rightarrow R_b > (\pi\lambda_b)^{-1/2} = 0.4$. Since $R_b > 0.6$ in the evaluation, the condition should be satisfied. The comparison between approximate hit probability and simulations are shown in Fig. 2(a). The curves exhibit a very good match. The evaluation shows that the independence approximation (CIA) works very well in this general model with PPs.

• **multi-LRU-All**: The total hit probability is evaluated numerically using (20). The characteristic time per cache is found by solving (19) using a fixed point method, where the individual hit probability of each object is given in (18).

We provide a method to estimate the surfaces $|\mathcal{A}_{m_o}|$, $m_o = 1, \ldots, M$ for the Boolean/PPP case: A user u_o has a distance $R_{d,i}$ from each one of the m_o nodes x_i that cover her/him. These distances are realisations of a random variable, whose expected value can be found equal to $\mathbb{E}[R_d] = 2R_b/3$, i.e. the user lies in expectation at $2R_b/3$ away from the center of each covering disc. Then we have:

i) The coverage cell size (for the Boolean model) is the disc surface, equal to $|\mathcal{A}_1| = |\mathcal{C}| = \pi R_b^2$.

ii) When $M \to \infty$, a disc having center the user u_o and radius $R_M = R_b + \mathbb{E}[R_d] = 5R_b/3$ is (due to randomness of node positions) fully covered. So $|\mathcal{A}_M| = |\mathcal{C}|(5/3)^2$.

iii) For intermediate cases $1 < m_o < M$, the surface should be somewhere between the two extremes, and obviously the surface $|\mathcal{A}_{m_o}|$ should be monotone increasing with m_o. We also expect that for low m_o, the total area $|\mathcal{A}_M|$ will be filling fast, whereas for larger ones, the change in surface should be small. For this we can use a function with exponential decrease for large m_o, such as

$$|\mathcal{A}_{m_o}| = |\mathcal{A}_M|(1 - e^{-m_o\delta}), \quad \delta = -\ln(1 - \tfrac{|\mathcal{A}_1|}{|\mathcal{A}_M|}). \quad (28)$$

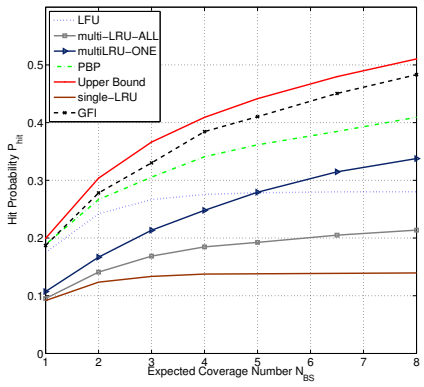
The comparison between approximate hit probability and simulations are shown in Fig. 2(b). The approximation and simulation curves seem to closely follow one another. For large values of the radius, the approximation curves seem to diverge from the simulations. This should be less a failure of the CSA approximation (which is shown to be accurate for the two-cache network in the Appendix), but more possibly a failure of the above method to approximate well the surfaces $|\mathcal{A}_{m_o}|$. More accurate values of $|\mathcal{A}_{m_o}|$ should exhibit a better fit.

2) Comparison of policies: Hit versus Coverage Number.

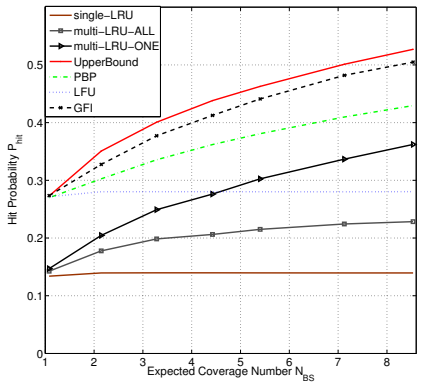
In Fig. 3(a), 3(b) and 3(c) we evaluate the hit probabilities of the proposed *multi-LRU* policies over the expected number of covering stations. In the simulations the radius of the Boolean model is increased from $R_b = 0.6$ to 2.25. The radius is mapped to the expected coverage number, as in Table II. In Fig. 3(a) transmission nodes are positioned as a PPP, while in Fig. 3(b), 3(c) on a Lattice. We compare the multi-LRU-One/All performance with different existing policies mentioned in this paper, like LFU, single-LRU, PBP [7] and GFI [19], as well as the upper bound given in (2). The parameter α is chosen equal to 1% in Fig. 3(a), 3(b) and 5% in Fig. 3(c). In both $F = 10000$, $\gamma = 0.78$.

As a reminder, the single-LRU policy is not influenced by multi-coverage. Each user can contact a single station, the one closest to the user. If the user request is cached in this memory, then there is a hit, otherwise the object is fetched from the core network and inserted to the station's cache.

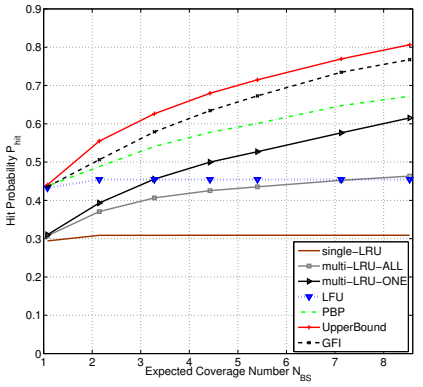
From the three figures very interesting conclusions about the policies can be derived:



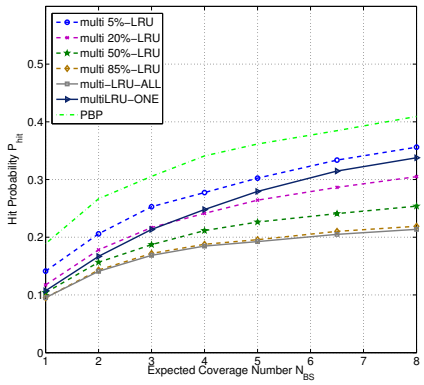
(a) Hit Probability PPP/Boolean, $\alpha = 1\%$.



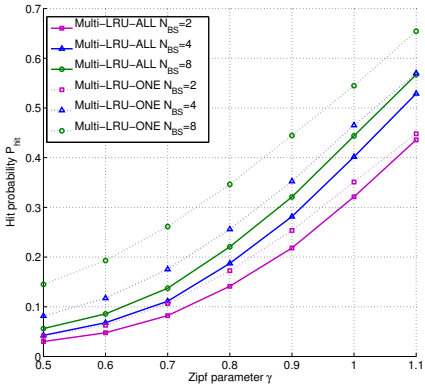
(b) Hit Probability Lattice/Boolean, $\alpha = 1\%$.



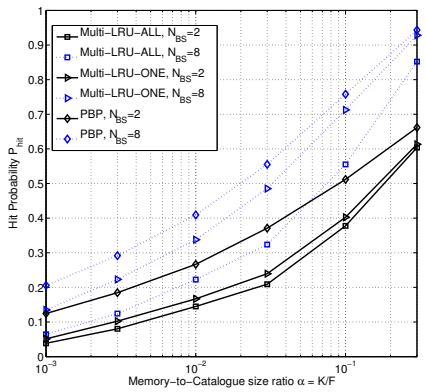
(c) Hit Probability Lattice/Boolean, $\alpha = 5\%$.



(d) q-multi-LRU-All, Boolean/PPP, $\alpha = 1\%$.



(e) Hit Probability over γ, PPP, $\alpha = 1\%$.



(f) Hit Probability over ρ, PPP, $\gamma = 0.78$.

Fig. 3. IRM traffic. Hit Probability evaluation of the two multi-LRU policies and comparison with existing POP and POQ policies.

(i) Even for small values of coverage overlap (expected coverage number) a considerable increase in hit probability is achieved by using the multi-LRU policies, compared to the single-LRU. For $\alpha = 1\%$, when $\overline{N_{BS}} = 2$, the multi-LRU-One is 42% (relative gain) above the single-LRU for Lattice placement and 35% for PPP placement. A further increase of $\overline{N_{BS}}$ makes the gain even more apparent. For $\overline{N_{BS}} = 3$ the relative gains are 70% and 60%, respectively.

(ii) For every value of $\overline{N_{BS}}$ the multi-LRU-One policy performs better that the multi-LRU-All, in all three figures. This is because, for stationary traffic, when the same object is inserted in all stations covering a user (case -All), a request falling in areas of overlap profits less by the diversity of content from the multiple covering stations.

(iii) From both figures, the difference in performance between POP {LFU, PBP, GFI} and POQ {multi-LRU-One/All, simple LRU} policies is evident. For IRM traffic, POP policies have greater performance by exploiting the "expensive" information of known object popularity, which is assumed constant over time. In a realistic environment however, where traffic patterns change with time, such policies will demand regular updates and are approximative, because they depend on estimations over the popularity values. On the other hand, the multi-LRU policies introduced here do not depend on such information. A related interesting remark is that, as the MCSR α increases, the difference between the two groups' performance decreases. This can be observed by comparing Fig. 3(c) to Fig. 3(b) (Lattice).

(iv) For $\overline{N_{BS}}$ close to 1, a user can connect to approximately one station, and the performance of multi-LRU-One/All, and single-LRU coincide. The same applies for the group LFU, PBP and GFI. For $\overline{N_{BS}} \approx 1$ these last three policies tend to cache the K most popular objects in each station. Hence, when a user connects to a single station then she/he gets the maximum hit probability and the upper bound also coincides.

(v) It is obvious that the two standard policies single-LRU and LFU exhibit constant performance as the multi coverage event increases, because the memory of each station is updated independently of the others and a user is served by at most one station.

(vi) GFI performs best among all policies, and its performance is very close to the upper bound. The latter is an indication that the upper bound is fairly tight. The good performance of the GFI comes at the cost of very high computational complexity for the memory allocations, as well as centralised implementation that requests considerable amount of information availability.

3) q-LRU: Fig. 3(d) plots the hit probability of q-multi-LRU-All policies for various values of $q \in (0, 1]$. As in the previous figures, $\gamma = 0.78$, $F = 10,000$, $\lambda_b = 0.5$ and stations are modelled by PPP and have memory $K = 100$.

When $q = 1 = 100\%$, q-multi-LRU-All reduces to the

multi-LRU-All policy. As q decreases, the performance of q-multi-LRU-All improves, but new content is inserted more rarely. In this sense, the good performance of q-multi-LRU-All with small q exploits the IRM characteristic of stationary traffic, and will converge to good performance after a long transient period. This is often not realistic for traffic that exhibits faster variations in popularity and catalogue content.

4) Zipf parameter γ: We provide plots for the hit probability versus this parameter in Fig. 3(e). Letting γ increase results in a popularity distribution where a small number of objects is considerably more popular than the rest of the catalogue. Eventually, hit probability will increase for both multi-LRU policies, because due to the Update phase, popular objects tend to be kept cached in memory once inserted and get hit more often. Furthermore, the relative difference $\frac{P_{hit}(multi-LRU-One)-P_{hit}(multi-LRU-All)}{P_{hit}(multi-LRU-All)}$ decreases as γ increases. This happens because for increasing γ unpopular objects have less influence on the hit probability.

5) Memory-to-Catalogue-Size Ratio MCSR α: Fig. 3(f) illustrates the behaviour of the three policies {multi-LRU-One, multi-LRU-All, PBP} when varying α (here a larger size catalogue of $F = 20,000$ is used, in order to evaluate for very small values of the $\alpha = K/F$ ratio). The hit probability increases when the ratio α increases, and tends to 100% as the ratio tends to 1. Furthermore, the need for smart memory allocations is less important for large values of the ratio $\alpha = K/F$ because the sum popularity of files left outside the caches is not considerable. Thus, we reasonably see in the figure, that different policies tend to have the same performance for larger values of the ratio α.

B. Traffic with temporal locality

The evaluation up to this point has been restricted to IRM user (request) traffic, and it was observed in the simulation figures that the multi-LRU-One performs better than the multi-LRU-All. This is because IRM is stationary, so, by letting the simulations run for a long time period the performance of the multi-LRU-One can converge to high hit probabilities. This however is generally not true for traffic that exhibits temporal locality, like the model we introduce in Section V of our work (see also [38], [31]).

For the performance evaluation we consider $N_{st} = 20$ stations placed on a square lattice in a rectangular window, and the Boolean coverage model. If not mentioned otherwise, $\lambda_c = 2400$ [objects/day], with expected request volume per content $\mathbb{E}[V] = 2.1$. The distribution lifespan p.d.f. is truncated Pareto, with $T \in [\tau_{min}, \tau_{max}] = [0.1, 96]$ $[days]$, and $\mathbb{E}[T] = 35$. The overall duration of the simulations is 15 $[months]$. The popularity shape g_i of content c_i is chosen with probability $a_1 = 0.06$ as Logistic, $a_2 = 0.38$ as Gompertz and with $a_3 = 0.56$ as Exponential. Later we include the Uniform shape.

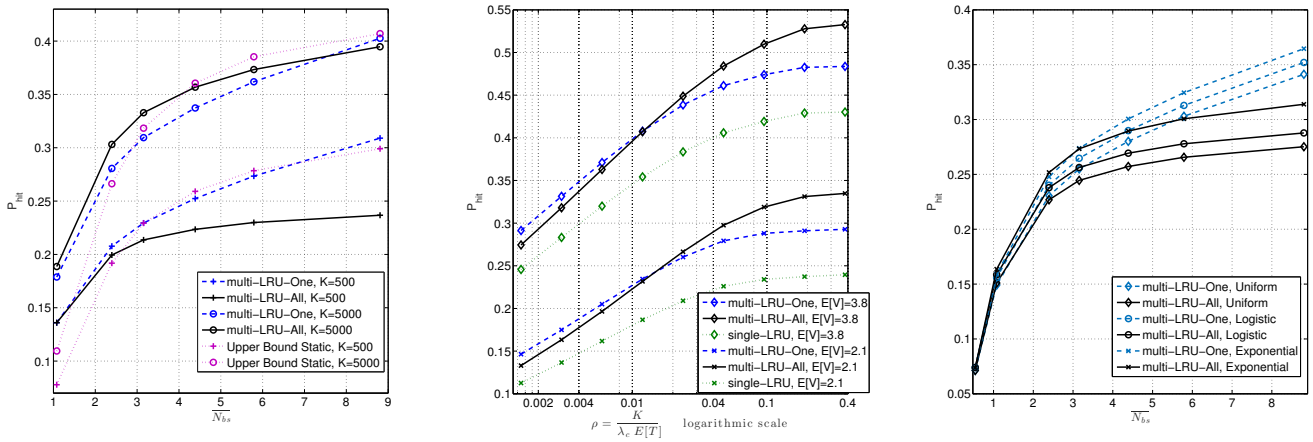
In each simulation, we keep some variables fixed and let others vary, to evaluate the policy performance and understand their influence. The variables to be varied are categorised into (1) *network* variables (mean coverage number $\overline{N_{bs}}$, cache size K), and (2) *traffic* variables (average request volume per content $\mathbb{E}[V]$, mean lifespan $\mathbb{E}[T]$, probability vector for the shape (a_1, a_2, a_3)).

1) Network influence: To increase the hit probability, data-objects should both: stay long-enough in each cache, and be inserted in as many caches as possible. But since storage space per station is limited, a trade-off arises, which is captured by the two variations of the multi-LRU policy (-One and -All). This trade-off is depicted in Fig. 4(a) for different values of the memory size. The larger the storage space, the longer it takes for an object to be evicted. So, for large memory the geographical expansion of an object is beneficial. This is shown for $K = 5000$ in which case multi-LRU-All surpasses multi-LRU-One. But for each K, there is a critical value of $\overline{N_{bs}}$ after which the performance of the -All is less than -One, because after this value further increase of content diversity is at the cost of content variety. The smaller the cache size, the more valuable storage space becomes because an object stays less time in the cache before eviction. Hence, for smaller K -One shows better performance and exceeds the -All variation even for small values of multi-coverage $\overline{N_{bs}}$.

In Fig. 4(b) the performance of multi-LRU policies versus the CCSR ratio ρ (26) is illustrated. This ratio is equal to the mean number of memory slots per active content and is a measure of the system's storage capability, because the smaller it is than one, the less storage resources are available. Keeping the denominator of ρ constant, Fig. 4(b) shows the impact of the memory size on the policy performance. Obviously, hit probability increases with K, but for smaller K, the -One variation is preferable to the -All, as explained also previously. There is again a critical point in K after which the -All variation is preferable (for large storage). If the ratio ρ is further increased, the performance gains are diminishing for both variations, and saturation occurs.

2) Traffic influence: The qualitative impact of the mean lifespan value on the hit probability can be understood by reading Fig. 4(b) in the opposite direction of the x-axis, from right to left. Keeping the numerator constant, as ρ decreases $\mathbb{E}[T]$ increases. This means that the same storage capacity serves a larger active catalogue size $\mathbb{E}[\mathcal{F}]$. Consequently the overall performance drops. Moreover, Fig. 4(b) illustrates that the hit probability improves as the average number of requests per content $\mathbb{E}[V]$ increases. The reason is that for higher $\mathbb{E}[V]$ a content put in storage is requested and hit more times.

In Fig. 4(c) each curve corresponds to a scenario where contents are assumed to follow only one particular popularity shape. Specifically, either the logistic, or the negative exponential, or the uniform shape is used. In the negative exponential shape popularity takes big values in a short time period and then drops abruptly. A steep popularity shape implies that consecutive requests of the same content appear close to each other in time. This makes more probable the event that the content is not evicted before its next request happens. With this in mind, it can be understood that the negative exponential can lead to higher hit probabilities than the uniform shape. Interestingly, for isolated caches the authors in [38], [31] state that the shape does not affect significantly the hit probability of LRU. In our model, this can be observed when $\overline{N_{bs}}$ takes small values so every user can connect to at most one station, and



(a) Hit Prob. VS $\overline{N_{bs}}$ (various cache-size K). (b) Hit Prob. VS CCSR ρ (various $\mathbb{E}[V]$). (c) Hit Prob. VS $\overline{N_{bs}}$ (different popularity shapes).

Fig. 4. Evaluation of the hit probability of multi-LRU policies for different system variables. Parameter values in (b) $\overline{N_{bs}} = 2.4$. In (c) $K = 1500$.

this observation is confirmed in Fig 4(c). But as $\overline{N_{bs}}$ increases and multi-coverage effects appear, *the multi-LRU performance depends considerably on the correlation between requests of the same content, and thus the shape of the popularity curves*.

3) Comparison with the single-LRU: Under single-LRU a user can access only one (the closest in this work) station's memory even when covered by more than one. As a result hit performance is independent of $\overline{N_{bs}}$ (provided coverage is enough so that a user is always covered by at least one station). Depriving the user of the ability to retrieve its content from all covering stations, strongly reduces the overall hit probability. In Fig. 4(b), where $\overline{N_{bs}} = 2.4$, both multi-LRU policies show relative gains compared to the single-LRU, for all values of ρ. The maximum gains reach 30% when $\mathbb{E}[V] = 2.1$ and 20% when $\mathbb{E}[V] = 3.8$.

4) Comparison with centralised Policies with periodic Popularity updates and prefetching (POP): Fig. 4(a) shows that even the upper bound in (27) for POP policies with estimated popularity input does not surpass the appropriate multi-LRU policy in performance, except maybe for a small range of $\overline{N_{bs}}$. For static IRM traffic, we saw in Fig.3(a)-3(c) that multi-LRU performed lower than the centralised policies POP. But, under a temporal traffic model (which is also more realistic), the ability of multi-LRU policies to update at each request the caches, without the need to estimate the content popularities, results in a considerable performance boost.

VII. Conclusions

In this work we have introduced a novel family of spatial multi-LRU policies, which exploit multi-coverage events of wireless networks to increase the hit probability. Two main variations are investigated, the multi-LRU-One and the -All. Che-like approximations give results close to simulation values. The multi-LRU-One provides higher object diversity in neighbouring caches and performs better under IRM traffic. The multi-LRU-All instead, lets objects quickly spread geographically and makes them immediately available to many users. This variation is profitable for traffic with temporal locality. Hence, depending on the incoming traffic either policy can be recommended. Future work should explain more clearly how the performance of these policies is affected by the spatial and temporal locality characteristics of traffic.

Acknowledgment

The authors kindly thank Jim Roberts (IRT-SystemX, INRIA) for helpful discussions that improved the current work.

References

[1] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira. Characterizing reference locality in the WWW. *IEEE PDIS*, 1996.

[2] A. Araldo, D. Rossi, and F. Martignon. Design and evaluation of cost-aware information centric routers. *ACM SIGCOMM ICN*, pages 147–156, 2014.

[3] A. Avranas, A. Giovanidis. Performance of spatial Multi-LRU caching under traffic with temporal locality. *9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, 2016.

[4] F. Baccelli and B. Błaszczyszyn. *Stochastic Geometry and Wireless Networks, Volume I — Theory*, volume 3, No 3–4 of *Foundations and Trends in Networking*. NoW Publishers, 2009.

[5] F. Baccelli and B. Błaszczyszyn. On a coverage process ranging from the Boolean model to the Poisson Voronoi tessellation, with applications to wireless communications. *Adv. in Appl. Probab.*, (SGSA) 33, pp. 293-323, 2001.

[6] E. Bastug, M. Bennis, M. Kountouris, and M. Debbah. Cache-enabled small cell networks: Modeling and Tradeoffs. *EURASIP J. Wireless Comm. and Networking*, 41, 2015.

[7] B. Błaszczyszyn and A. Giovanidis. Optimal geographic caching in cellular networks. *IEEE ICC*, 2015.

[8] B. Błaszczyszyn and H. P. Keeler. Equivalence and comparison of heterogeneous cellular networks. *PIMRC/WDN-CN*, 2013.

[9] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. *INFOCOM*, 2010.

[10] A. Brodersen, S. Scellato, and M. Wattenhofer. YouTube around the world: geographic popularity of videos. *21st WWW*, 2012.

[11] H. Che, Y. Tung, and Z. Wang. Hierarchical web caching systems: Modeling, design and experimental results. *IEEE JSAC*, 20(7):1305–1314, Sept. 2002.

[12] E. Coffman and P. Denning. *Operating Systems Theory*. Englewood Cliffs (NJ): Prentice-Hall, 1973.

[13] A. Dan and D. Towsley. An approximate analysis of the LRU and FIFO buffer replacement schemes. *ACM SIGMETRICS*, 1990.

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

[15] S. E. Elayoubi and J. Roberts. Performance and cost effectiveness of caching in the mobile access network. *ACM SIGCOMM ICN*, 2015.

[16] R. Fagin, T.G. Price. Efficient calculation of expected miss ratios in the independent reference model. *SIAM J. Comput.*, 7:288–297, 1978.

[17] S. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, and S. Shenker. Less pain, most of the gain: Incrementally deployable ICN. *SIGCOMM Comput. Commun. Rev.*, 43(4):147–158, Aug. 2013.

[18] C. Fricker, P. Robert, and J. Roberts. A versatile and accurate approximation for LRU cache performance. *24th International Teletraffic Congress*, 2012.

[19] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire. Femtocaching: Wireless video content delivery through distributed caching helpers. *INFOCOM*, 2012.

[20] A. Giovanidis, A. Avranas. Spatial Multi-LRU Caching for Wireless Networks with Coverage Overlaps. *ACM SIGMETRICS '16*, Antibes Juan-les-Pins, France, 2016.

[21] M. Haenggi. *Stochastic Geometry for Wireless Networks*. Cambridge University Press, 2013.

[22] B. A. Huberman and L. A. Adamic. The nature of markets in the World Wide Web. *Computing in Economics and Finance*, 521, 1999.

[23] P. Jelenković and A. Radovanović. Least-recently-used caching with dependent requests. *Elsevier Theoretical Computer Science*, pages 293–327, 2004.

[24] S. Jin and A. Bestavros. Sources and characteristics of web temporal locality. *IEEE MASCOTS*, 2000.

[25] H. P. Keeler, B. Błaszczyszyn, and M. K. Karray. SINR-based k-coverage probability in cellular networks with arbitrary shadowing. In *IEEE ISIT*, 2013.

[26] M. Leconte, M. Lelarge, and L. Massoulié. Designing adaptive replication schemes in distributed content delivery networks. *27th International Teletraffic Congress*, 2015.

[27] E. Leonardi, and G.L. Torrisi. Least recently used caches under the Shot Noise Model. *INFOCOM*, 2015.

[28] V. Martina, M. Garetto, and E. Leonardi. A unified approach to the performance analysis of caching systems. *INFOCOM*, 2014.

[29] K. Naveen, L. Massoulie, E. Baccelli, A. C. Viana, and D. Towsley. On the interaction between Content Caching and Request Assignment in Cellular Cache Networks. *AllThingsCellular'15*, 2015.

[30] M. E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46:323–351, 2005.

[31] F. Olmos, B. Kauffmann, A. Simonian, and Y. Carlinet. Catalog dynamics: Impact of content publishing and perishing on the performance of a LRU cache. *26th International Teletraffic Congress*, 2014.

[32] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah. Wireless Caching: Technical Misconceptions and Business Barriers. *IEEE Communications Magazine*, 54, no. 8, pp.16-22, 2016.

[33] K. Poularakis, G. Iosifidis, and L. Tassiulas. Approximation algorithms for mobile data caching in small cell networks. *IEEE Trans. on Communications*, 62(10), October 2014.

[34] C. Richier, E. Altman, R. Elazouzi, T. Altman, G. Linares, and Y. Portilla. Modelling View-count Dynamics in YouTube. *arXiv:1404.2570*.

[35] V. Sourlas, P. Flegkas, G.S. Paschos, D. Katsaros, and L. Tassiulas. Storage planning and replica assignment in content-centric publish/subscribe networks. *Elsevier Computer Networks*, 55(18):4021–4032, Dec. 2011.

[36] R. Timo, and M. Wigger. Joint cache-channel coding over erasure broadcast channels. *IEEE ISWCS*, Brussels, Belgium, 2015.

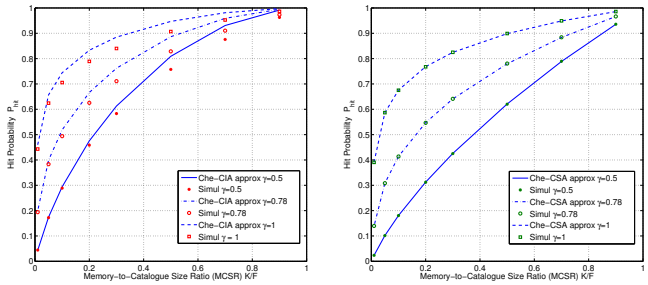
[37] S. Tamoor-ul Hassan, M. Bennis, P. H. Nardelli, and M. Latva-Aho. Modeling and analysis of content caching in wireless small cell networks. *IEEE ISWCS*, 2015.

[38] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini. Unravelling the impact of temporal and geographic locality in content caching systems. *IEEE Trans. on Multimedia*, 17(10):1839–1854, 2015.

[39] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. Polyzos. A survey of Information-Centric Networking. *IEEE Communications Surveys & Tutorials*, 16(2):1024 – 1049, 2014. Second Quarter.

APPENDIX

To understand how the Che-like approximations work for the multi-LRU policies, we analyse the simple network of two nodes x_i, $i \in \{1,2\}$, each one equipped with a cache of size K. Each node covers an entire area $A \subset \mathbb{R}^2$, so that all planar points are covered by both nodes. The total area is divided in two Voronoi cells $\mathcal{V}(x_i)$. To simplify further, we assume equal-sized Voronoi cells $|\mathcal{V}(x_1)| = |\mathcal{V}(x_2)| = |\mathcal{V}|$.



(a) multi-LRU-One (b) multi-LRU-All

Fig. 5. Che approximations for (a) multi-LRU-One with CIA and (b) multi-LRU-All with CSA, in the two-cache network. Hit probability versus MCSR $a = K/F$, $F = 10,000$ objects, for different Zipf parameter γ.

We apply the analysis of Section IV-B to this network model. Specifically, the formula for the hit probability of an object c_j at cache Ξ_i in (7), takes the expression (for $i = \{1,2\}$),

$$P_{hit,i}(j) \overset{IRM}{=} \mathbb{P}(u_{-1} \in (\mathcal{S}_{-1}, |t_o - t_{-1}| < T_C, j)) \cdot \cdot [P_{hit,i}(j) + \mathbb{P}(c_j \notin \Xi_1 \cap c_j \notin \Xi_2)] \quad (29)$$

Solving the above over $P_{hit,i}(j)$ gives an expression for the hit probability of object c_j at cache Ξ_i. The characteristic time T_C is found by solving the equation (8),

$$\sum_{j=1}^{F} P_{hit,i}(j) = K, \quad i = \{1,2\}. \quad (30)$$

Finally, the total hit probability (9) takes both caches into account, and is equal to

$$P_{hit} = \sum_{j=1}^{F} a_j \left(1 - \mathbb{P}(c_j \notin \Xi_1 \cap c_j \notin \Xi_2)\right). \quad (31)$$

• **multi-LRU-One**: In this case, $\mathcal{S}_o = \mathcal{S}_{-1} = \mathcal{V}(x_i)$, in (29). Table III gives all pairs of inventory states that a user u_o arriving at t_o^- sees, when the previous user u_{-1} asking for the same content arrived in cell (say) $\psi_{-1} \in \mathcal{V}(x_1)$ at some time t_{-1}^-, such that $|t_o - t_{-1}| \leq T_C$. We denote by logical 1 the fact that the object is in the cache and by 0 otherwise. From the table it is clear that user u_{-1} does not take any action on cache Ξ_2, this is why, when $\mathbf{1}\left[c_j \in \Xi_2(t_{-1}^-)\right] = 1$, we cannot know whether the content will remain in the cache till t_o^-, so we write $\mathbf{1}\left[c_j \in \Xi_2(t_o^-)\right] \in \{0,1\}$.

TABLE III
MULTI-LRU-ONE: STATES AT t_{-1}^- AND t_o^-

$\Xi_1(t_{-1}^-)$	$\Xi_2(t_{-1}^-)$		$\Xi_1(t_o^-)$	$\Xi_2(t_o^-)$	
0	0	$\rightarrow$	1	0	insert 1
0	1	$\rightarrow$	0	$\{0,1\}$	no update
1	0	$\rightarrow$	1	0	update 1
1	1	$\rightarrow$	1	$\{0,1\}$	update 1

There is the unknown probability $\mathbb{P}(c_j \neq \Xi_1 \cap c_j \neq \Xi_2) = 1 - \mathbb{P}(c_j \in \Xi_1 \cup c_j \in \Xi_2)$. For multi-LRU-One, we observe

that an insertion of an object is triggered when its request arrives but does not find the object inside any of the two caches. However, the insertion is done only in the closest cache and stays there for time T_C. During this time, the same object cannot be inserted in the other cache, hence, $\{c_j \in \Xi_1\}$ and $\{c_j \in \Xi_2\}$ are mutually exclusive events. Then,

$$\begin{aligned}\mathbb{P}\left(c_j \notin \Xi_1, c_j \notin \Xi_2\right) &= 1 - \mathbb{P}\left(c_j \in \Xi_1 \cup c_j \in \Xi_2\right) \\ &= 1 - 2P_{hit,1}(j), \end{aligned} \quad (32)$$

where the last equality is due to the symmetry of our model and the IRM traffic. However, in more general cases of node placement and coverage, content exclusivity is not true, because only a small area of the coverage cell will overlap with one neighbour. Users in other areas of the cell will be covered by other neighbours that can trigger the insertion of the same object, anyway. Hence, this result is not of much use for the PP coverage models. For this reason we want to evaluate how the CIA approximation applies here. For the two-cache model, this means for Ξ_1 (or Ξ_2),

$$\mathbb{P}\left(c_j \notin \Xi_1 \cap c_j \notin \Xi_2\right) = 1 - P_{hit,1}(j), \quad (CIA_1). \quad (33)$$

We can then replace in (29) and (30) to get (for $i \in \{1, 2\}$)

$$P_{hit,i}(j) = 1 - e^{-a_j \lambda_u |\mathcal{V}| T_C}, \quad (34)$$

$$\sum_{j=1}^{F} P_{hit,i}(j) = \sum_{j=1}^{F} \left(1 - e^{-a_j \lambda_u |\mathcal{V}| T_C}\right) = K. \quad (35)$$

For the total P_{hit} probability, we should appropriately adapt the form in (31) to the CIA_2 approximation,

$$\begin{aligned} P_{hit} &= \sum_{j=1}^{F} a_j \left(1 - \left(\mathbb{P}\left(c_j \notin \Xi_1\right)\right)^2\right) \\ &= \sum_{j=1}^{F} a_j (1 - e^{-a_j \lambda_u 2|\mathcal{V}| T_C}), \end{aligned} \quad (36)$$

and the area $2|\mathcal{V}| = |A|$ is equal to the total coverage cell.

• **multi-LRU-All**: In this case, $\mathcal{S}_{-1} = \mathcal{S}_o = A$ in (29) , for the hit probability of node i.

To calculate the unknown probability $\mathbb{P}\left(c_j \notin \Xi_1 \cap c_j \notin \Xi_2\right)$ we argue as follows. In the case of multi-LRU-All, an object cannot be inserted in cache 1 if not inserted also in cache 2 and the other way round. Based on the Che approximation, once the object is inserted it stays T_C amount of time, before removed from each cache. Hence, the existence of an object in one cache implies the existence of the same object in the other. So, due to the model's symmetry

$$\begin{aligned}\mathbb{P}\left(c_j \notin \Xi_1 \cap c_j \notin \Xi_2\right) &= 1 - \mathbb{P}\left(c_j \in \Xi_1 \cup c_j \in \Xi_2\right) \\ &= 1 - P_{hit,1}(j). \end{aligned} \quad (37)$$

This is simply the *Cache Similarity Approximation (CSA)*, which for the two-cache network is exact! Then (29) gives,

$$P_{hit,i}(j) = 1 - e^{-a_j \lambda_u |A| T_C}. \quad (38)$$

To find the characteristic time, we need to solve (30),

$$\sum_{j=1}^{F} P_{hit,i}(j) = \sum_{j=1}^{F} \left(1 - e^{-a_j \lambda_u |A| T_C}\right) = K. \quad (39)$$

The total hit probability is equal to,

$$\begin{aligned} P_{hit} &= \sum_{j=1}^{F} a_j \left(1 - \mathbb{P}\left(c_j \notin \Xi_1, c_j \notin \Xi_2\right)\right) \\ &\overset{(38)}{=} \sum_{j=1}^{F} a_j (1 - e^{-a_j \lambda_u |A| T_C}). \end{aligned} \quad (40)$$

An *alternative way* to calculate $\mathbb{P}\left(c_j \notin \Xi_1(t_o), c_j \notin \Xi_2(t_o)\right)$ is the following. A user u_o finds the two caches without object c_j, if the previous user u_{-1} (at say $\mathcal{S}_{-1} = \mathcal{V}(x_1)$) with the same demand, arrived either (i) at $t_{-1}^- : |t_o - t_{-1}| > T_C$, so that whatever the state of the two caches $\Xi_1(t_{-1}^-)$, $\Xi_2(t_{-1}^-)$, the object c_j is eventually removed, since more than T_C elapsed till t_o, or (ii) at $|t_o - t_{-1}| \leq T_C$. In the second case all possible change of states for the two caches is shown in Table IV. From this, we note that, the object will always be found in at least one of the two caches at t_o, so that the time difference can not be smaller than T_C. Hence,

$$\mathbb{P}\left(c_j \notin \Xi_1(t_o), c_j \notin \Xi_2(t_o)\right) \overset{IRM}{=} e^{-a_j \lambda_u |A| T_C}. \quad (41)$$

The expressions in (41) and (38) are the same.

TABLE IV
MULTI-LRU-ALL: STATES AT t_{-1}^- AND t_o^-

$\Xi_1(t_{-1}^-)$	$\Xi_2(t_{-1}^-)$		$\Xi_1(t_o^-)$	$\Xi_2(t_o^-)$	
0	0	$\rightarrow$	1	1	insert both
0	1	$\rightarrow$	0	1	update 2
1	0	$\rightarrow$	1	0	update 1
1	1	$\rightarrow$	1	1	update both

The accuracy of the approximations in the two-cache network is shown in Fig.5. The Che-CIA approximation for multi-LRU-One - although not accurate - performs reasonably well in the two-cache network. The Che-CSA approximation for the multi-LRU-All, is exact.