Soft Cache Hits and the Impact of Alternative Content Recommendations on Mobile Edge Caching[*]

Thrasyvoulos Spyropoulos
Dept. Mobile Communications
EURECOM,France
spyropou@eurecom.fr

Pavlos Sermpezis
Institute of Computer Science
FORTH, Greece
sermpezis@ics.forth.gr

ABSTRACT

Caching popular content at the edge of future mobile networks has been widely considered in order to alleviate the impact of the data tsunami on both the access and backhaul networks. A number of interesting techniques have been proposed, including femto-caching and "delayed" or opportunistic cache access. Nevertheless, the majority of these approaches suffer from the rather limited storage capacity of the edge caches, compared to the tremendous and rapidly increasing size of the Internet content catalog. We propose to depart from the assumption of hard cache misses, common in most existing works, and consider "soft" cache misses, where if the original content is not available, an alternative content that is locally cached can be recommended. Given that Internet content consumption is increasingly entertainment-oriented, we believe that a related content could often lead to complete or at least partial user satisfaction, without the need to retrieve the original content over expensive links. In this paper, we formulate the problem of optimal edge caching with soft cache hits, in the context of delayed access, and analyze the expected gains. We then show using synthetic and real datasets of related video contents that promising caching gains could be achieved in practice.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Store and forward networks, Wireless communication; C.4 [**Performance of Systems**]: Modelling techniques

Keywords

Caching; Opportunistic networks; Mobile data offloading; Optimization; Recommendation Systems

1. INTRODUCTION

In the context of cellular networks, it is widely believed that aggressive densification, overlaying the standard macro-cell network with a large number of small cells (e.g., pico- or femto-cells), is a promising way of dealing with the ongoing data crunch [1]. As this densification puts a tremendous pressure on the backhaul network, researchers have suggested storing popular content at the "edge", e.g., at small cells [2], user devices [3, 4, 5, 6], or vehicles acting as mobile relays [7] in order to avoid congesting the capacity-limited backhaul links, and reduce the access latency to such content.

Local content caching has been identified as one of the five most disruptive enablers for 5G networks [8], sparking a tremendous interest of academia and industry alike. While caching had been widely studied in peer-to-peer systems and content distribution networks (CDNs) [9], the number of storage points required in future dense HetNets are many orders of magnitude more than in traditional CDNs (e.g., 1000s small cells per area covered by one CDN server). Therefore, the storage space per local cache must be significantly smaller to keep costs reasonable. Hence, even though studies assuming a large (CDN-type) cache deep inside the core network [10] give promising hit ratios, only a tiny fraction of the constantly and exponentially increasing content catalog could realistically be stored at each edge, leading to low "local" cache hit ratios [11, 12].

Additional "global" caching gains could be sought by increasing the "effective" cache size visible to each user through: (a) small cell overlaps, where each user is in range of multiple cells and caches (e.g., in the femto-caching case [2]), (b) collocated users overhearing the same broadcast channel and benefiting from cached content in other users' caches (as in coded caching [13]), and (c) delayed content access, where a user might wait up to a TTL for its request, during which time more than one (fixed [6] or mobile [4, 5, 7]) caches can be seen. These ideas could theoretically increase the cache hit ratio significantly, when the "global" cache size becomes large enough (e.g., when, in the latter example, the aggregate size of all caches a user sees within a TTL becomes comparable to the content catalog). Nevertheless, in most practical cases a local edge cache would realistically fit at most $10^{-3}/10^{-4}$ of the catalog (e.g., just the entire Netflix catalogue is about 3PBs). Even if the above methods offered a $10\times$ effective cache increase, they would not suffice to achieve significant cache hit ratios (e.g., in the notation of [13], the key factor KM/N would be equal to 10^{-2}, leading to a global caching gain of $\frac{1}{1+10^{-2}}$, a mere 1% of extra gain).

Operators, are thus left with a very costly dilemma: bear a huge cost for the backhaul infrastructure (e.g., fiber everywhere) or bear a huge cost for CDN-like storage at each and every small cell. We believe this dilemma stems from the common underlying assumption of almost every caching scheme to try to satisfy *every* possible user request, either from the local cache or, in the worst case, the content server. This leads to an immense catalogue of potential content. Our main assertion in this paper is that, in an Internet

[*]This work was partially funded by the French Government (National Research Agency, ANR) through the "Investments for the Future" Program reference #ANR-11-LABX-0031-01.

CHANTS'16, October 03-07, 2016, New York City, NY, USA
© 2016 ACM. ISBN 978-1-4503-4256-8/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2979683.2979688

which is becoming increasingly content-centric and entertainment-oriented, a radically different approach could be beneficial, namely *moving away from satisfying a given user request towards satisfying the user.* E.g., a user requesting a content X, not available locally (e.g., a fan wanting to follow last weekend's premier league's games), might be equally satisfied (in the best case) or not fully dissatisfied (in many cases), if she receives another content Y related to X (e.g., another premier league game from that weekend). Another example is users streaming content *in sequence* (e.g., browsing YouTube videos back-to-back or listening to personalized radio). In that case, the selected content at each step is often *recommended related to the previous one*, and the user might be almost equally happy with many alternatives. We will use the term *soft cache hit* to describe such scenarios. Finally, we believe such a system is timely given the recent interest of content providers with sophisticated recommendation engines, such as NetFlix and YouTube (i.e., Google), to act as Mobile Virtual Network Operators (MVNO) in the context of RAN Sharing [14].

To this end, we perform here a preliminary analysis and performance evaluation of such a system, in order to obtain initial insights. We first formulate the problem of edge caching with *soft cache hits*, and analyze the expected gains. We then show using both synthetic data and a real dataset of related video contents that interesting caching gains could be achieved in practice. Our problem formulation and analysis takes place in the context of *delayed content access* via static or mobile small cells [7, 6], for two reasons: (a) we believe such delayed access is interesting for low-cost users (e.g., 2 euro plans for operators like Free [15]) or developing regions, and (b) could be easily combined with soft cache hits to achieve multiplicative gains. Nevertheless, the basic tenets of our approach are equally applicable to femto-caching (i.e., the framework of [2]) or even other PHY-aware caching systems [16].

To the best of our knowledge, the closest related work to the idea of soft cache hits is Roadcast [17], proposing a query-response based P2P VANET system, where users' query requirements can be relaxed in order to get a matching response sooner. Nevertheless, this work focuses mostly on content similarity metrics and considers heuristics to achieve a square root based allocation policy, known to be optimal in P2P systems. Square root policies are suboptimal in our problem setup, as proven later, with or without soft cache hits [7].

2. PROBLEM SETUP

Content Model: We consider a wireless network with randomly distributed users, requesting contents from a catalogue $\mathcal{K}$ with $\|\mathcal{K}\| = K$ contents. A user requests content $i \in \mathcal{K}$ with probability p_i. Without loss of generality ("w.l.o.g.") we assume all contents have the same size.

Network Model: Our network consists of M small cells (SC). These SCs can be either static (as in the femto-caching model [2]) or mobile (e.g. a vehicular cloud as in [7]). We denote the set of all SCs as $\mathcal{M}$. We also assume that each SC is equipped with storage capacity of C contents. Accessing content directly from the local cache, i.e. a *cache hit*, is considered "cheap" while a *cache miss* leads to an "expensive" access (e.g. of the backhaul link in [2] or the macro-cell in [7]).

Delayed Access Protocol: If the requested content is not available in a nearby small cell, the user waits until it encounters other small cells (as a result of user or cell mobility), until a Time-To-Live T. If the content is not found in any SC within T, a cache miss occurs and the content is fetched over the expensive link.

Meeting Model: Meetings between each user and each SC are IID, with the *residual* time until such a meeting occurs being a random variable with CDF $F(t)$.

LEMMA 2.1. *If there are N total SCs storing the requested content, the probability of not encountering any of them within T is*

$$P_{miss}(N) = \overline{F(T)}^{N} \tag{1}$$

The above result follows directly from the definition of $F(t)$ and the assumption of IID meetings.

For simplicity, in this paper we will focus on $F(t) = 1 - \exp^{-\lambda t}$, so that $P_{miss}(N) = \exp^{-\lambda N t}$. The identical meeting rates assumption can be further relaxed, as explained in Section 5.

Up to this point, the problem setup is the same as in [7, 6]. The main departure from that model is captured in the following.

Content Relation Graph: Each content $i \in \mathcal{K}$ has a set of *related contents*. Let u_{ij} denote the utility a given user gets if she originally asks for content i but instead receives content j, where $0 \leq u_{ij} \leq 1$ and $u_{ii} = 1, \forall i$. The set of related contents $\mathcal{R}_i \subseteq \mathcal{K}$ can be formally defined as: $\mathcal{R}_i = \{j \in \mathcal{K} : j \neq i, u_{ij} > 0\}$. These relations define a content relation matrix (or graph) $\mathbf{U} = \{u_{ij}\}$.

Delayed Access with Soft Cache Hits (SCH): A user again performs delayed access. However, if the requested content i is not found within T, but a content in $j \in \mathcal{R}_i$ is found in one of the encountered caches, a soft cache hit occurs (and thus no expensive access is needed). A cache miss occurs if neither the requested nor any related content is found within T, in which case the original content is retrieved over the expensive link. The soft cache hit utility is equal to u_{ij}. We will consider two main cases for $\mathbf{U}$.

- *Soft Cache Hits (Case 1):* $u_{ij} = 1, \forall j \in \mathcal{R}_i$. Any related content gives a cache hit. As soon as one is found, the user stops looking.
- *Soft Cache Hits (Case 2):* $u_{ij} = c \ (0 < c < 1), \forall j \in \mathcal{R}_i$. If a related content $j \in \mathcal{R}_i$ is found before T, the user now continues looking for i until T. If it fails, a *soft cache hit* occurs and the access to the expensive link is still avoided. However, the utility attained is less than 1 (equal to c), which creates an interesting tradeoff. If neither i nor any related j is found by T, then a cache miss occurs, as usual.

3. CACHING WITH RELATED CONTENT

3.1 Objectives

The goal in the above defined problem is to minimize the number of bytes accessed over the expensive "link" (which is, as explained, a radio access link to a macro-cell and/or the backhaul network). When all contents have the same size, this is simplified to minimizing the number of (expensive) accesses, or equivalently, *maximizing the cache hit ratio*.

DEFINITION 1 (FEASIBLE PLACEMENT). *Let N_i denote the number of SC caches storing content i. A placement vector $\mathbf{N} = \{N_1, \ldots, N_K\}$ is "feasible", if it satisfies the following constraints:*

$$0 \leq \quad N_i \quad \leq M, \tag{2}$$

$$\sum_{i=1}^{K} N_i \quad \leq \quad M \cdot C. \tag{3}$$

N_i are the main optimization variables for our problem. Constraint (2) says that the number of SCs storing content i is non-negative and at most equal to the total number of SCs, and constraint (3) that the total number of content replicas stored at all the edge caches cannot exceed their total capacity.

In the traditional case of delayed access no soft cache hits are allowed. This will serve as our *baseline* scenario. The problem objective (i.e., the expected hit ratio) in this case is given in the following lemma.

LEMMA 3.1 (CACHE HIT RATIO - BASE). *Assume a feasible placement vector* $\mathbf{N}$*. The cache hit rate, i.e., the expected number of user requests served locally when no soft cache hits are allowed is equal to*

$$g_{Base}(\mathbf{N}) = \sum_{i=1}^{K} p_i \cdot \left(1 - e^{-\lambda \cdot T \cdot N_i}\right). \tag{4}$$

The objective (Eq.(4)) in the above lemma is straightforward in light of Lemma 2.1 and the model of Section 2.

As explained earlier, when we do allow soft cache hits, if content i is requested, a cache hit can occur also if other contents j (related to i) can be accessed on time. The modified objective for Cases 1 and 2 of the content relation graph $\mathbf{U}$ is given in the following two lemmas (the proofs are based on basic probabilistic arguments, and are omitted for brevity).

LEMMA 3.2 (SOFT CACHE HIT RATIO (CASE 1)). *Assume a feasible placement vector* $\mathbf{N}$*, and a content relation graph* $\mathbf{U}$*, where* $u_{ij} \in \{0,1\}, \forall i,j \in \mathcal{K}$*. The cache hit rate for* $\mathbf{N}$ *is equal to*

$$g_{SCH1}(\mathbf{N}) = \sum_{i=1}^{K} p_i \cdot \left(1 - e^{-\lambda \cdot T \cdot \sum_{j=1}^{K} N_j \cdot u_{ij}}\right) \tag{5}$$

LEMMA 3.3 (SOFT CACHE HIT RATIO (CASE 2)). *Assume a feasible placement vector* $\mathbf{N}$*, and a content relation graph* $\mathbf{U}$*, where* $u_{ii} = 1, \forall i$*, and* $u_{ij} \in \{0, c\}, \forall j \in \mathcal{K} \backslash \{i\}$*. The cache hit rate for* $\mathbf{N}$ *is equal to*

$$\begin{aligned} g_{SCH2}(\mathbf{N}) = \sum_{i=1}^{K} p_i \cdot \Big[&\left(1 - e^{-\lambda \cdot T \cdot N_i}\right) \\ &+ c \cdot e^{-\lambda \cdot T \cdot N_i} \cdot \left(1 - e^{-\lambda \cdot T \cdot \sum_{j \in R_i} N_j}\right) \Big] \end{aligned} \tag{6}$$

The main difference between these two cases is that, in the first case, finding a related content gives utility 1 and is equivalent to a normal cache hit. However, in the second case, a related content allows the operator to avoid accessing the expensive link, but is penalized because the utility for the user is lower, leading to a utility of $c < 1$ (we remind the reader that $\mathcal{R}_i$ in the second term of Eq.(6) includes all related contents j, such that $u_{ij} > 0$, but does not include content i).

3.2 Performance Improvement Under the Baseline Placement

Maximizing the objective of Lemma 3.1 within the feasibility region of Definition 1, defines the optimal cache allocation problem for the baseline scenario (no soft cache hits). This is in general an INLP (Integer Non-Linear Program) that relates to a "multiple knapsack" problem (with equal capacities and logarithmic rather than linear utilities) and is NP-hard to solve. Various polynomial approximation algorithms exist with good performance when the size of the caches are large enough to fit many contents. One such approximation can be achieved by solving a continuous relaxation of the problem (related to the fractional knapsack problem), where the optimization variables $N_i \in [0, M]$ are continuous. In that case, it is easy to show that the baseline problem is convex, whose optimal solution can be found analytically using Lagrangian multipliers and solving the KKT conditions (we refer the interested reader to [7] for more details). Specifically, the optimal solution is given by

$$N_i^* = \begin{cases} 0, & \text{if } p_i < L \\ \frac{1}{\lambda T} \ln\left(\frac{p_i \lambda T}{\rho}\right), & \text{if } L \leq p_i \leq U \\ M, & \text{if } p_i > U \end{cases} \tag{7}$$

where $L \triangleq \rho \cdot (\lambda T)^{-1}$, $U \triangleq \rho \cdot (\lambda T)^{-1} \cdot e^{\lambda \cdot M \cdot T}$, and ρ is an appropriate Lagrange multiplier corresponding to the capacity constraint of Eq.(3).[1]

Replacing N_i^* in the objective of the baseline problem (Eq.(4)) gives us the optimal cache hit ratio, if we ignored related content. At the same time, replacing N_i^* in the objective of Eq.(5) gives us the cache hit ratio when we can satisfy a request with related content, *but the caching decisions were already taken and are the original ones*. (We will show later that we could do even better by considering the related content graph $\mathbf{U}$ when solving the cache placement problem.) The following theorem provides the expected improvement in terms of load on the expensive link, for a simple scenario where $L \leq p_i \leq U, \forall i \in \mathcal{K}$.

THEOREM 3.4. *Assume that* $\|\mathcal{R}_i\| = L, \forall i \in \mathcal{K}$*. The expected improvement in the cache hit ratio by recommending alternative contents, when the optimal cache placement algorithm is oblivious to these recommendations, is equal to*

$$\frac{1 - g_{Base}(\mathbf{N}^*)}{1 - g_{SCH1}(\mathbf{N}^*)} = K \cdot \left(\frac{\lambda T}{\rho}\right)^{L-1} \frac{1}{\sum_{i \in \mathcal{K}} p_i \cdot \Pi_{j \in \mathcal{K}} \frac{1}{p_j^{u_{ij}}}} \tag{8}$$

PROOF. The cache miss ratio (or "load" on the main infrastructure) in the baseline problem is $1 - g_{Base}(\mathbf{N}^*)$. Replacing Eq.(7) into Eq.(4) gives

$$\begin{aligned} 1 - g_{Base}(\mathbf{N}^*) &\overset{Eq.(4)}{=} \sum_{i=1}^{K} p_i \cdot e^{-\lambda \cdot T \sum_{j=1}^{K} \cdot N_i^*} \\ &\overset{Eq.(7)}{=} \sum_{i=1}^{K} p_i \cdot e^{\ln\left(\frac{\rho}{p_i \lambda T}\right)} = \sum_{i=1}^{K} p_i \cdot \frac{\rho}{p_i \lambda T} = \frac{K\rho}{\lambda T} \end{aligned} \tag{9}$$

Similarly, let's assume that an original request could be satisfied with a related content as in Lemma 3.2. The cache miss ratio, denoted as $1 - g_{SCH1}(\mathbf{N}^*)$, can be calculated as:

$$\begin{aligned} 1 - g_{SCH1}(\mathbf{N}^*) &\overset{Eq.(5)}{=} \sum_{i=1}^{K} p_i \cdot e^{-\lambda \cdot T \cdot \sum_{j=1}^{K} N_j^* \cdot u_{ij}} \\ &\overset{Eq.(7)}{=} \sum_{i=1}^{K} p_i \cdot e^{-\lambda \cdot T \cdot \left(\sum_{j=1}^{K} \frac{1}{\lambda T} \ln\left(\frac{p_j \lambda T}{\rho}\right) \cdot u_{ij}\right)} \\ &= \sum_{i=1}^{K} p_i \cdot e^{\sum_{j=1}^{K} \ln\left(\frac{\rho}{p_j \lambda T}\right) \cdot u_{ij}} \\ &= \sum_{i=1}^{K} p_i \cdot \Pi_{j \in \mathcal{K}} \left(\frac{\rho}{\lambda T} \frac{1}{p_j} \cdot u_{ij}\right) = \left(\frac{\rho}{\lambda T}\right)^{L} \sum_{i=1}^{K} p_i \cdot \Pi_{j \in \mathcal{K}} \frac{1}{p_j^{u_{ij}}} \end{aligned}$$

Hence, the gain from soft cache hits (case 1) is equal to $\frac{1 - g_{Base}(\mathbf{N}^*)}{1 - g_{SCH1}(\mathbf{N}^*)}$, which gives the desired Eq.(4). □

[1] An integer solution could be obtained by rounding [7, 2]. Alternatively, one could interpret a non-integer N_i value as follows: If $N_i = 7.6$, 100% of content i is allocated to 7 caches, and one more cache stores only 60% of the content. If a user encounters the latter, she retrieves the remaining 40% from the infrastructure.

The case where some contents receive no or maximum (M) copies, as in Eq.(7), can be easily derived by modifying the summation in the above proofs. As a very simple example, consider the case of uniform content popularity, i.e. $p_i = \frac{1}{K}$. After some simple calculations, we get that the performance benefits by related content are equal to $\left(\frac{K\rho}{\lambda T}\right)^{-(L-1)}$. However, we know that $\frac{K\rho}{\lambda T} \leq 1$, since it is the cache miss rate of the base policy (see Eq.(9)). Therefore, the above gain $\left(\frac{K\rho}{\lambda T}\right)^{-(L-1)} \geq 1$, and is increasing in $L-1$, the number of related contents per content i, as one would expect. A similar result can be easily derived for Case 2, as well as when the number of non-zero elements on each row i of $\mathbf{U}$ is different (i.e. not all equal to L).

3.3 Content Graph Aware Optimal Caching

We have so far assumed that the caching policy is unaffected by the ability to recommend alternative contents. While this already leads to performance gains, as shown earlier, it is still suboptimal. For example, assume a user requesting content A would be OK to receive instead content B (i.e. $u_{AB} = 1$) and a user requesting content B would be OK to receive content A instead (i.e. $u_{BA} = 1$). If both contents A and B are popular, *a standard caching policy would give a high number of replicas to both*, according to Eq.(7). However, this is clearly suboptimal here, since the caching algorithm could just store only one of the two at each cache, saving valuable capacity that could be used to store other contents. The following two theorems formalize this for the two content relation graph cases, discussed in Section 2. Due to space limitations, we only show the proof for the more generic Case 2.

THEOREM 3.5 ($\mathbf{U}$-AWARE OPTIMAL CACHING (CASE 1)). *Assume a content relation graph* $\mathbf{U}$, *where* $u_{ij} \in \{0,1\}, \forall i,j \in \mathcal{K}$. *The optimal content placement that directly exploits related contents is given by vector* $\mathbf{N^*_{SCH1}}$ *which is the solution to the following optimization problem*

$$\underset{\mathbf{N}}{\textit{maximize}} = \sum_{i=1}^{K} p_i \cdot \left(1 - e^{-\lambda \cdot T \cdot \sum_{j=1}^{K} N_j \cdot u_{ij}}\right),$$

$$\textit{subject to } \mathbf{N} \textit{ feasible (according to Definition 1)}$$

Furthermore, the above problem is a convex optimization problem.

THEOREM 3.6 ($\mathbf{U}$-AWARE OPTIMAL CACHING (CASE 2)). *The optimal content placement defined by maximizing the objective of Lemma 3.3, subject to the feasibility constraints of Definition 1, gives the optimal content allocation vector* $\mathbf{N^*_{SCH2}}$. *Furthermore, the problem is also convex.*

PROOF. It is easy to see that the feasibility region (Definition 1) is convex. The objective function needs to be concave (since this is formulated as a maximization problem). A sufficient condition is if its Hessian matrix $\mathbf{H}$ is negative semi-definite, i.e., $\mathbf{z}^T \cdot \mathbf{H} \cdot \mathbf{z} \leq 0$, $\forall \mathbf{z} = \{z_i\} \geq 0$.

Taking the derivatives of the objective function g_{SCH2}, we calculate the terms of the Hessian matrix

$$H_{m,m} = -(\lambda \cdot T)^2 \cdot \Big[p_m \cdot (1-c) \cdot e^{-\lambda \cdot T \cdot N_m} + \sum_{i=1}^{K} p_i \cdot c \cdot \mathcal{I}_{im} \cdot \mathcal{I}_{in} \cdot e^{-\lambda \cdot T \cdot \sum_{j=1}^{K} N_j \cdot \mathcal{I}_{ij}}\Big]$$

and for $m \neq n$

$$H_{m,n} = -(\lambda \cdot T)^2 \sum_{i=1}^{K} p_i \cdot c \cdot \mathcal{I}_{im} \cdot \mathcal{I}_{in} \cdot e^{-\lambda \cdot T \cdot \sum_{j=1}^{K} N_j \cdot \mathcal{I}_{ij}}$$

where $\mathcal{I}_{nm}$ is 1 if $u_{nm} > 0$; otherwise is 0.

Then, the product $\mathbf{z}^T \cdot \mathbf{H} \cdot \mathbf{z}$ is given by the expression

$$\mathbf{z}^T \cdot \mathbf{H} \cdot \mathbf{z} = -(\lambda \cdot T)^2 \sum_{m=1}^{K} \Big[z_m^2 \cdot p_m \cdot (1-c) \cdot e^{-\lambda \cdot T \cdot N_m} + \sum_{n=1}^{K} \sum_{i=1}^{K} z_m \cdot z_n \cdot p_i \cdot \mathcal{I}_{im} \cdot \mathcal{I}_{in} \cdot e^{-\lambda \cdot T \cdot \sum_{j=1}^{K} N_j \cdot \mathcal{I}_{ij}}\Big]$$

which is always ≤ 0. □

4. PERFORMANCE EVALUATION

4.1 Simulations Setup

Mobility Trace. We use the TVCM mobility model to generate a trace, where nodes move in a square area $1000m \times 1000m$ comprising three sub-areas of interest (communities). Each node moves inside its community for 60% of the time, and leaves it for a few short periods. The area is entirely covered by macro-cell BSs, and also includes 25 non overlapping small-cell base stations (SCs), with a communication range of 100m.

Content Popularity. We create $K = 1000$ contents and assign to each of them a popularity value p_i drawn from a Zipf distribution, $p_i \in [1, 1000]$ with shape parameter $\alpha = 2$. Power-law distributions have been shown to capture well real popularity patterns [18, 19, 20].

Utility Matrix. To investigate the effect of the matrix U, we generate different matrices belonging to two generic classes:
(a) random U: for each content pair $\{i, j\}$, the utility is $u_{ij} = 1$ with probability $p = \frac{L}{K}$ (otherwise it is 0), such that each content has on average L related contents, i.e., $L = E[||\mathcal{R}_i||]$.
(b) popularity proportional U: for each content pair $\{i, j\}$, the utility is $u_{ij} = 1$ with probability $p = L' \cdot \frac{\cdot p_j}{\sum_j p_j}$ (otherwise it is 0), where p_j is the popularity of content j, and L' is a normalization parameter that determines $E[||\mathcal{R}_i||]$.

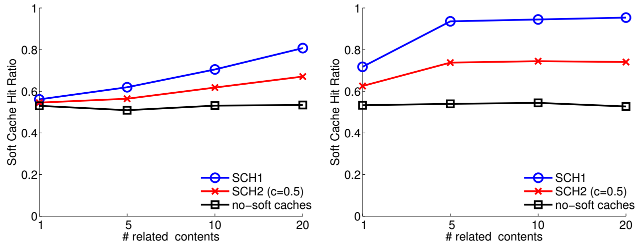
YouTube datasets. In addition to the synthetic popularity/utility patterns, we use real datasets from YouTube that contain information about *video popularity* and *related video lists* [21]. Table 1 contains information about the datasets we use, and some main statistics. We pre-process the data to remove entries with 0 or no popularity value. For each video j appearing in the related videos list of a video i, we set $u_{ij} = 1$ and $u_{ji} = 1$. Due to the sparseness of the datasets, we consider only the videos belonging to the largest connected component of the graph with vertices $\mathcal{V} = \{i : i \in \mathcal{K}\}$ and edges $\mathcal{E} = \{\epsilon_{ij} : i, j \in \mathcal{K}, u_{ij} = 1\}$.

4.2 Effects of Utility Matrix

We first study the effects of the (a) density, $L = E[||\mathcal{R}_i||]$, and (b) type (*random U / popularity proportional U*) of the utility matrix. Specifically, in Fig. 1 we present the soft-cache hit ratio for the *SCH1* and *SCH2* (with $c = 0.5$) cases, under the base optimal policy $\mathbf{N}^*$, as well as the hit ratio of the scenarios without soft caches (*no-soft caches*). As expected, the cache hit rate improves as the density L of the matrix U (x-axis) increases. Under random U matrices the increase in the cache hit rate is almost linear (Fig 1(a)) on L, but quickly plateaus for the popularity proportional U matrices

Table 1: YouTube dataset instances information (after processing).

	Data (date / depth of search [21])	K	$E[\|\|\mathcal{R}_i\|\|]$
Instance 1	27 July 2008 / 3	2098	5.3
Instance 2	27 March 2008 / 1	1086	7.9



(a) random U (b) popularity proportional U

Figure 1: Scenarios where the matrices U are generated in (a) random and (b) popularity proportional way, so that the expected number of related contents per content equals the value of the x-axis. $Q = 20$ and $TTL = 5min$

(Fig. 1(b)). This is reasonable as the achieved cache hit ratios for the popularity proportional U case already reach values $> 90\%$, for few related contents. The reason is that popular contents that have higher probability to appear in the related list of other contents, are also stored in more caches (under the base optimal policy).

These initial observations show that the performance can be improved by recommending more contents (density) and/or by selecting carefully which contents to recommend (type of matrix U). This is a positive message, since there are more than one degrees of freedom for a system design, allowing thus improvements under various settings (e.g., restriction on the max number of recommended contents, predefined content relations), and enabling cross-layer (application/network) design and optimization approaches.

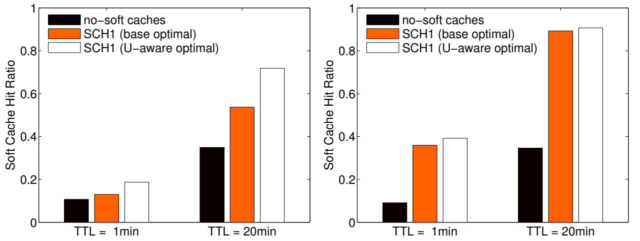
4.3 Gains of Optimal Caching Policies

In Fig. 2 we compare the performance gains of the base optimal policy $\mathbf{N}^*$ and the U-aware optimal policy $\mathbf{N}^*_{\mathbf{SCH1}}$ the *SCH1* case (Theorem 3.5). Under random U matrices (Fig. 2(a)), the achieved cache hit rate by $\mathbf{N}^*_{\mathbf{SCH1}}$ is always higher than in the $\mathbf{N}^*$ policy, with an increase of 44% (for $TTL =$ 1min) and 34% (for $TTL =$ 20min). Here, we need to stress that the extra performance gain from the U-aware optimal caching policy $\mathbf{N}^*_{\mathbf{SCH1}}$ comes without any cost for the system: the recommendation system (matrix U) and the caching capacity (M and Q) remain the same, and only the caching policy changes (i.e., in practice, this corresponds to a simple modification in the content placement algorithm).

In the popularity proportional U case (Fig. 2(b)), the performance improvement of the U-aware optimal policy $\mathbf{N}^*_{\mathbf{SCH1}}$ over the base optimal policy $\mathbf{N}^*$ is moderate (9% and 16%, for $TTL =$ 1min and $TTL =$ 20min, respectively). This indicates that when a recommendation system is carefully designed for a mobile environment (i.e., in our example, resulting to a popularity proportional matrix U), the U-aware caching policy does not add significant gains. As a result, only the content popularities is needed for the caching placement algorithm. Hence, the *network provider* does not need to cooperate further with a *content provider* (which designs also the recommendation system), e.g., YouTube or Netflix, and this facilitates the deployment of a soft-cache system in practice.

4.4 Gains of the YouTube's Recommendation System

We conduct simulations on the TVCM mobility trace using the popularity/utility patterns of the YouTube datasets (see Section 4.1). In Table 2 we present the relative gain in the soft-cache hit ratio, i.e., $g_{Base}(\mathbf{N}^*)$ vs. hit ratio under no-soft caches scenario. The improvement in performance by using soft-caches can be up to 20%, and -on average- is higher in *Instance 1* where the content catalogue is larger (cf. Table 1). The gains are similar in other simulated scenarios we tested; with parameters $Q = \{5, 10, 20, 50\}$ and $TTL = \{0.5, 1, 5, 20\}min$.



(a) random U (b) popularity proportional U

Figure 2: Scenarios where the matrices U are generated in (a) random and (b) popularity proportional way, so that the expected number of related contents per content is $L = 5$. The capacity of caches is $Q = 5$ (i.e., 0.5% of the catalogue size M).

Placing contents with the U-aware optimal policy $\mathbf{N}^*_{\mathbf{SCH1}}$ gives similar gains as in the $\mathbf{N}^*$ case in the simulated scenarios. In light of the synthetic results, this perhaps suggests that the content relation graph for these YouTube instances more closely resemble the popularity proportional case, rather than the random.

Table 2: Gains in cache hit ratio in the YouTube scenarios.

	Instance 1		Instance 2	
	$Q = 5$	$Q = 50$	$Q = 5$	$Q = 50$
$TTL = 1min$	11%	17%	12%	13%
$TTL = 20min$	20%	19%	11%	7%

5. DISCUSSION

Our initial results suggest that soft cache hits could be a promising way to make edge caching scale, opening up new interesting operator-user performance tradeoffs. Some limitations and potential extensions of the proposed model are discussed here.

User-dependent recommendations: Throughout this work, we have been assuming that the related contents for a requested content item i, and their related utilities depend only on item i, and not on the user that requested it. In a sense, this relates to *item-item* collaborative filtering, where a new/alternative item is recommended based on its similarity with the requested one. Item-item recommendations have been claimed to offer some advantages compared to *user-user* collaborative filtering [22]. Nevertheless, one user might be less happy than another, with the same alternative content. On the modeling side, one could take this into account by making u_{ij} a random variable and using its expected value $E[u_{ij}]$ in the objective functions of Section 3. Finally, on the recommendation side, a recommendation system could actually combine both types of collaborative filtering to make better recommendation. This would lead to different $\mathbf{U}$ graphs per user (or user clusters), whose integration and impact on our framework is part of future work.

Generalization of $\mathbf{U}$ *graph:* For simplicity, in our analysis we assumed that related contents bring the same amount of utility (1 in case 1, and $c < 1$ in case 2). In general, different related contents might bring different amounts of utility. We could generalize our model by assuming a *Case 3* where $u_{ii} = 1, u_{ij} \in [0, 1)$ $i \neq j$. As in Case 2, if a user requesting content i, accesses (before i) any content $j \in \mathcal{R}_i$, she will be satisfied $u_{ij} \in (0,1)$ (less than 1). She will keep on requesting i till time T, but will *not* accept any other related content[2]. Contrary to Case 2, however, the value of the utility u_{ij} (to be contributed at the objective function) is not known a priori, since we cannot know a priori which content $j \in R_i$ will be accessed. One can still derive a closed form objective function with appropriate conditioning on all possible j, but we defer elaborating on this scenario for future work.

Generic mobility: Although it would be quite hard to relax the independent mobility assumption (using traces in simulations, where most such assumptions break, tends to be the de facto way of testing this) the identical contact rate assumption could be relaxed. E.g., in the context of exponential meetings, it has been shown that heterogeneous rates could be approximated with their mean, either asymptotically or as a bound [23].

Soft Cache Hits for Femto-caching: The proposed approach of soft cache hits and alternative content recommendations could apply equally well to more traditional caching frameworks that do not allow any delay, as is the popular femto-caching framework [2]. The relation between users and small cells that each user can access is captured by a bipartite graph, and the control variables x_{kj} define whether a content k is stored in a cache j. In the case of $\mathbf{U}$ as in Case 1, if some user requests content i, and the small cells in her range are $\mathcal{G} \subseteq \mathcal{M}$, the hit probability is given by

$$1 - \Pi_{j \in \mathcal{G}} \cdot \Pi_{k=1}^{K} \left(1 - x_{kj}\right)^{u_{ik}}, \tag{10}$$

instead of $1 - \Pi_{j \in \mathcal{G}} \left(1 - x_{ij}\right)$, in the original femtocaching case (see [2] for more details).

6. CONCLUSIONS

In this paper, we have proposed the idea of *soft cache hits* for mobile edge caching systems with delay tolerance, where a user request can sometimes be (partially) satisfied, even if the original content is not available locally, by recommending some related contents. We have formulated and analyzed the performance of such a joint system, and derived the optimal related content aware cache placement. Our theoretical analysis and initial evaluation suggest that significant performance gains can be achieved, even with simple modifications to the baseline system. Furthermore, our results suggest that the structure of the content relation graph plays an important role on the actual achievable performance.

7. REFERENCES

[1] N. Sapountzis, T. Spyropoulos, N. Nikaein, and U. Salim, “Optimal downlink and uplink user association in backhaul-limited hetnetsn,” in *Proc. IEEE Infocom*, 2016.

[2] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless video content delivery through distributed caching helpers,” in *Proc. IEEE INFOCOM*, 2012.

[3] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, “Scaling Behavior for Device-to-Device Communications With Distributed Caching,” *IEEE Trans. Information Theory*, 2014.

[4] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, “Mobile data offloading through opportunistic communications and social participation,” *IEEE Trans. on Mobile Computing*, vol. 11, no. 5, 2012.

[5] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan, “Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays,” in *Proc. IEEE WoWMoM*, 2011.

[6] P. Sermpezis, , and T. Spyropoulos, “"effects of content popularity in the performance of content-centric opportunistic networking: An analytical approach and applications,” *ACM/IEEE Trans. on Networking*, 2016.

[7] L. Vigneri, T. Spyropoulos, and C. Barakat, “Storage on Wheels: Offloading Popular Contents Through a Vehicular Cloud,” in *Proc. IEEE WoWMoM*, 2016.

[8] F. Boccardi, R. Heath, A. Lozano, T. Marzetta, and P. Popovski, “Five Disruptive Technology Directions for 5G,” *IEEE Comm. Mag. SI on 5G Prospects and Challenges*, 2014.

[9] S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in *Proc. IEEE INFOCOM*, pp. 1–9, 2010.

[10] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, “To cache or not to cache: The 3g case,” *IEEE Internet Computing*.

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

[12] G. S. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, “Wireless caching: Technical misconceptions and business barriers,” *http://arxiv.org/abs/1602.00173*.

[13] M. A. Maddah-Ali and U. Niesen, “Fundamental limits of caching,” *IEEE Transactions on Information Theory*, vol. 60, pp. 2856–2867, May 2014.

[14] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, “Cellslice: Cellular wireless resource slicing for active ran sharing,” in *COMSNETS*, pp. 1–6, 2014.

[15] “Free mobile plans.” http://mobile.free.fr/.

[16] W. C. Ao and K. Psounis, “Distributed caching and small cell cooperation for fast content delivery,” in *Proc. ACM MobiHoc*, 2015.

[17] Y. Zhang, J. Zhao, and G. Cao, “Roadcast: A popularity aware content sharing scheme in vanets,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, pp. 1–14, Mar. 2010.

[18] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, “Youtube traffic characterization: A view from the edge,” in *Proc. ACM IMC*, 2007.

[19] J. Erman, A. Gerber, K. K. Ramadrishnan, S. Sen, and O. Spatscheck, “Over the top video: The gorilla in cellular networks,” in *Proc. ACM IMC*, 2011.

[20] P. Sermpezis and T. Spyropoulos, “Inferring content-centric traffic for opportunistic networking from geo-location social networks,” in *Proc. IEEE WoWMoM (AOC workshop)*, 2015.

[21] http://netsg.cs.sfu.ca/youtubedata/.

[22] G. Bresler, D. Shah, and L. Voloch, “Collaborative filtering with low regret,” in *Proc. ACM SIGMETRICS*, 2016.

[23] P. Sermpezis and T. Spyropoulos, “Delay analysis of epidemic schemes in sparse and dense heterogeneous contact environments,” Tech. Rep. http://www.eurecom.fr/~sermpezi/TechRep_HetEpid.pdf, Eurecom, 2012.

[2] An alternative approach would be to keep requesting every cache encountered for potentially better related content. However, we believe this might put a high burden on the battery of the UE and the UE-SC traffic.