Femto-Caching with Soft Cache Hits: Improving Performance with Related Content Recommendation

Pavlos Sermpezis[1], Thrasyvoulos Spyropoulos[2], Luigi Vigneri[2], and Theodoros Giannakas[2]
[1] ICS-FORTH, Greece, sermpezis@ics.forth.gr
[2] EURECOM, France, first.last@eurecom.fr

***Abstract*—Pushing popular content to cheap "helper" nodes (e.g., small cells with local storage) during off-peak hours has recently been proposed to cope with the increase in mobile data traffic. If the requested content is available locally at a helper node, both user and operator performance could benefit. Nevertheless, the collective storage of a few nearby helper nodes does not usually suffice to achieve a high hit rate in practice. In this paper, we investigate the concept of "soft cache hits" where, if the original content is not available, some locally cached related contents can be recommended. Given that Internet content consumption is entertainment-oriented, we argue that there exist scenarios where a user might accept an alternative content (e.g., better download rate for alternative content, low rate plans), thus avoiding to access expensive/congested links. We formulate the problem of optimal edge caching with soft cache hits in a sufficiently generic setup, propose an efficient algorithm, 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.**

I. Introduction

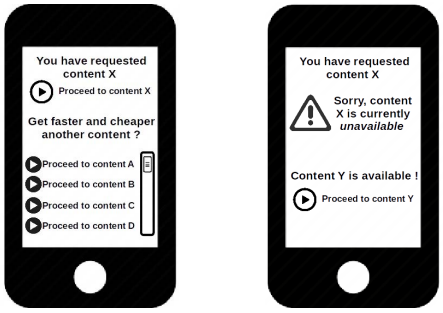
Mobile edge caching has been identified as one of the five most disruptive enablers for 5G networks [1], both to reduce content access latency and to alleviate backhaul congestion. However, the number of required storage points in future cellular networks will be orders of magnitude more than in traditional CDNs [2] (e.g., 100s or 1000s of small cells (SCs) corresponding to an area covered by a single CDN server today). As a result, the storage space per local edge cache must be significantly smaller to keep costs reasonable. Even if we considered a small subset of the entire Internet catalogue, e.g., a typical torrent catalogue (1,5 PB) or the Netflix catalogue (3 PB), cache hit ratios would still be low even with a relatively skewed popularity distribution and more than 1 TB of local storage [3], [4].

Additional caching gains have been sought by researchers, increasing the "effective" cache size visible to each user. This could be achieved by: (a) *Coverage overlaps*, where each user is in range of multiple cells, thus having access to the aggregate storage capacity of these cells, as in the femto-caching framework [5], [6]. (b) *Coded caching*, where collocated users overhearing the same broadcast channel may benefit from cached content in other users' caches [7]. (c) *Delayed content access*, where a user might wait up to a TTL for her request, during which time more than one cache (fixed [8] or mobile [9], [10], [11], [12]) can be encountered. Each of these ideas could theoretically increase the cache hit ratio (sometimes significantly), but the actual practical gains might not suffice by themselves, e.g., due to high enough cell density required for (a), sub-packetization complexity in (b), and imposed delays in (c).

We argue that, in an Internet which is becoming increasingly entertainment-oriented, *moving away from satisfying a given user request towards satisfying the user* could prove beneficial for caching systems. When a user requests a content not available in the local cache(s), a recommendation system could propose a set of *related content* that *is* locally available. If the user accepts one of these contents, an expensive remote access could be avoided. We will use the term *soft cache hit* to describe such scenarios.

There are a number of scenarios where soft cache hits are worth considering, as they could benefit both the user and the operator. (i) A *cache-aware recommendation* plugin to an existing application (e.g., the YouTube app) could, for example, let a user know that accessing the original content X is only possible at low quality (e.g., 240p) while related contents $A, B, C, ...$ could be streamed at high resolution, as shown in Fig. 1(a). (ii) Even more seamlessly for a user quality of experience, after she watches a clip X, the recommendation system could re-order its list of recommendations (among related contents of roughly equal similarity to X) to favor a cache hit in the next request [13], [14]. (iii) The operator can give incentives to users to accept the alternative contents when there is congestion (e.g., *zero-rating* services [15], [16]). (iv) In some cases, the operator might even "enforce" an alternative (but related) content (see Fig. 1(b)), e.g., offering low rate plans with higher data quotas with the agreement that, during congestion, only locally cached content can be served. (v) A cache-aware recommendation system could be used to improve service in "challenged networks" for developing areas [17] or when access to only a few Internet services is provided, e.g., the Facebook's Internet.org project [18].

Last but not least, we believe such a system is timely, given the increased convergence of Mobile Network Operators (MNO) and content providers with sophisticated recommendation engines (e.g., NetFlix and YouTube), in the context of Mobile Edge Computing (MEC) [19] and RAN Sharing [20]. E.g., a YouTube server application could be running at a MEC server collocated with the base station, together with a cache-aware recommendation component that has access to the (also local) cache content. Overall, the idea of soft cache hits is complementary and can be applied *on top* of existing proposals for edge caching, like the ones described earlier. In a recent



(a) (b)

Fig. 1: Mobile app examples for related content recommendation and *Soft Cache Hits*.

preliminary work [21], we have considered the idea of soft cache hits in a DTN context with mobile relays. Our goal in this paper is to develop the idea of soft cache hits in detail, applying it to standard mobile edge caching systems with cache cooperation (e.g., [5]). To our best knowledge, this is the first work to jointly consider related content recommendation gains and cache cooperation (e.g., femto-caching) gains. In this context, our main contributions are:

- *Soft Cache Hits (SCH) model:* We propose a generic model for mobile edge caching and alternative soft cache hits that can capture a number of interesting scenarios (Section II).
- *Femto-caching with SCH:* We formulate the problem of femto-caching with SCH. We show that the problem is NP-hard, and propose an efficient approximation algorithm with provable performance (Section III).
- *Validation:* We show using both synthetic data and a real dataset of YouTube related videos that additional caching gains, e.g., *on top of what femto-caching provides*, could be achieved in practice (Section IV).

Finally, we discuss related work and future research directions in Section V, and conclude our paper in Section VI.

II. Problem Setup

A. Network and Caching Model

Network Model: Our network consists of a set of users $\mathcal{N}$ ($|\mathcal{N}| = N$) and a set of SCs (or, *helpers*) $\mathcal{M}$ ($|\mathcal{M}| = M$). Users are mobile and the SCs with which they associate might change over time. Since the caching decisions are taken in advance (e.g., the night before, as in [5], [6], or once per few hours or several minutes), it is hard to know the exact SC(s) each user will be associated at the time she requests a content. To capture user mobility, we propose a more generic model than the fixed bipartite graph of [5]:

$$q_{ij} \doteq Prob\{\text{user i in range of SC j}\},$$

or, equivalently, q_{ij} is the percentage of time a user i spends in the coverage of SC j. Hence, deterministic q_{ij} ($\in \{0,1\}$) captures the static setup of [5], while uniform q_{ij} ($q_{ij} = q, \forall i,j$) represents the other extreme (no advance knowledge).

Content Model: We assume each user requests a content from a catalogue $\mathcal{K}$ with $|\mathcal{K}| = K$ contents. A user $i \in \mathcal{N}$ requests content $k \in \mathcal{K}$ with probability p_k^i.[1]

Cache Model (Baseline): We assume that each SC/helper is equipped with storage capacity of C contents (all our proofs hold also for different cache sizes). We use the integer variable $x_{kj} \in \{0,1\}$ to denote if content k is stored in SC j[2]. In the traditional caching model (baseline model), if a user i requests a content k which is stored in some nearby SC, then the content can be accessed directly from the local cache and a *cache hit* occurs. This type of access is considered "cheap", while a *cache miss* leads to an "expensive" access (e.g., over the SC backhaul and core network).

For ease of reference, the notation is summarized in Table I.

TABLE I: Important Notation

$\mathcal{N}$	set of users ($\|\mathcal{N}\| = N$)
$\mathcal{M}$	set of SCs / helpers ($\|\mathcal{M}\| = M$)
C	storage capacity of a SC
q_{ij}	probability user i in range of SC j
$\mathcal{K}$	set of contents ($\|\mathcal{K}\| = K$)
p_k^i	probability user i to request content k
x_{kj}	content k is stored in SC j ($x_{kj} = 1$) or not ($x_{kj} = 0$)
u_{kn}^i	utility of content n for a user i requesting content k
$F_{kn}(x)$	distribution of utilities u_{kn}^i, $F_{kn}(x) = P\{u_{kn}^i \leq x\}$
u_{kn}	average utility for content pair $\{k,n\}$ (over all users)

B. Soft Cache Hits

Up to this point the above model describes a baseline setup similar to the popular femto-caching framework [5]. The main departure in this paper is the following.

Alternative Content Recommendation: When the requested content is not found in a local cache, a list of related contents (out of the ones already cached) is recommended to the user (see the example in Fig. 1(a)). If a user selects one of them, a *(soft) cache hit* occurs, otherwise there is a *cache miss* and the network must fetch and deliver the original content.[3] Whether a user accepts an alternative content or not depends both on the content (how related it is) and the user; we describe this behavior with the model of Definition 1.

Definition 1. *A user i that requests a content k that is not available, accepts a recommended content n with probability u_{kn}^i, where $0 \leq u_{kn}^i \leq 1$, and $u_{kk}^i = 1, \forall i,k$.*

The utilities/probabilities (in the remainder we use these terms interchangeably) define a content relation matrix $\mathbf{U^i} = \{u_{kn}^i\}$ for each user. Per user utilities u_{kn}^i could be estimated from past statistics and/or user profiles, as usually done by

[1] This generalizes the standard femto-caching model [5] which assumes same popularity per user. We can easily derive such a popularity p_k from p_k^i.

[2] Due to space limitations, we develop our theory assuming that all contents have the same size (e.g., content chunks); the extended framework for variable content sizes can be found in [22].

[3] Throughout our proofs, we assume, for simplicity, that the user can pick *any* of the available cached contents; however, our analysis holds also when only a small subset of locally cached contents is recommended (e.g., the ones the recommender thinks are the most related for that user/request, as in [14]).

standard recommendation algorithms [23]. In some cases, the system might have a coarser view of these utilities (e.g., item-item recommendation [24]). We develop our theory and results for the most generic case of Definition 1, but we occasionally refer to the following two subcases, which might appear in practice:

Sub-case 1: The system does not know the exact utility u_{kn}^i for each node i, but only how they are distributed among all nodes, i.e., the distributions $F_{kn}(x) \equiv P\{u_{kn}^i \leq x\}$.

Sub-case 2: The system knows only the *average utility* u_{kn} per content pair $\{k, n\}$.

III. Femtocaching with Soft Cache Hits

A request from a user i for a content $k \in \mathcal{K}$ would result in a (standard) cache hit only if at least one SC/helper j within range (i.e., $q_{ij} = 1$) stores content k in the cache (i.e., if $x_{kj} = 1$). Hence, the (baseline) *cache hit ratio* for this request can be expressed as a function of integer variables:

$$CHR(i,k) = 1 - \prod_{j=1}^{M} \left(1 - q_{ij} \cdot x_{kj}\right).$$

If we further allow for soft cache hits, the user might be also satisfied by receiving a different content $n \in \mathcal{K}$. The probability of this event is, by Definition 1, equal to u_{kn}^i. The following Lemma derives the total cache hit rate in that case.

Lemma 1 (Soft Cache Hit Ratio (SCHR))**.** *Let $SCHR$ denote the expected cache hit ratio for a user i requesting content k (including regular and soft cache hits). Then,*

$$SCHR(i,k,\mathbf{U^i}) = 1 - \prod_{j=1}^{M}\prod_{n=1}^{K}\left(1 - u_{kn}^i \cdot x_{nj} \cdot q_{ij}\right) \quad (1)$$

Proof. Consider a helper j and a user i requesting content k. If j is in range of i, the probability that user i could be satisfied with some related content n stored in j is $u_{kn}^i \cdot x_{nj}$, since u_{kn}^i gives the probability of acceptance (by definition), and x_{nj} denotes if content n is stored in the cache (note that this includes the case of a regular hit, if $n = k$). However, whether helper j is in range of i is captured by q_{ij}. Hence, $P\{n|k,i,j\} = u_{kn}^i \cdot x_{nj} \cdot q_{ij}$. Considering now all neighboring base stations j as above, as well as all related contents n gives Eq. (1). □

Lemma 1 can be easily modified for the the sub-cases 1 and 2 of Definition 1 presented in Section II-B. We state the needed changes in Corollary 2 (the proof can be found in [22]).

Corollary 2. *Lemma 1 holds for the the sub-cases 1 and 2 of Definition 1, by substituting in the expression of Eq.* (1) *the term u_{kn}^i with*

$$u_{kn}^i \to E[u_{kn}^i] \equiv \int_0^\infty \left(1 - F_{kn}(x)\right) dx \qquad \textit{(for sub-case 1)}$$

$$u_{kn}^i \to u_{kn} \qquad \textit{(for sub-case 2)}$$

Considering further (i) every user i in the system, (ii) all possible content requests, and their respective probabilities p_k^i, and (iii) the capacity constraint, gives us the following discrete optimization problem.

Problem 1. *The optimal cache placement problem for the femtocaching scenario with soft cache hits and content relations described by $\mathbf{U^i} = \{u_{kn}^i\}$, $\forall i \in \mathcal{N}$, is*

$$\underset{X=\{x_{11},\ldots,x_{KM}\}}{\textit{maximize}} \; f(X) =$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot \left(1 - \prod_{j=1}^{M}\prod_{n=1}^{K}\left(1 - u_{kn}^i \cdot x_{nj} \cdot q_{ij}\right)\right), \quad (2)$$

$$s.t. \quad \sum_{k=1}^{K} x_{kj} \leq C, \quad \forall j \in \mathcal{M}. \quad (3)$$

The following lemma gives the complexity of the above optimization problem.

Lemma 3. *Problem 1 is NP-hard, even for the case of a single helper.*

Proof. We will prove that the problem with soft cache hits is NP-hard already for a simple instance where each user is associated with only one helper (i.e., $\sum_j q_{ij} = 1, \forall i$). In that case, we can treat each cache independently, and drop the second index for both the storage variables ($x_{kj} \to x_k$) and connectivity variables ($q_{ij} \to q_i$). Let us further assume that the utilities are equal among all users and can be either 1 or 0, i.e., $u_{kn}^i = u_{kn},\ \forall i \in \mathcal{N}$ and $u_{kn} \in \{0,1\},\ \forall k,n \in \mathcal{K}$. We denote as $\mathcal{R}_k$ the set of contents related to content k, i.e.

$$\mathcal{R}_k = \{n \in \mathcal{K} : n \neq k, u_{kn} > 0\} \text{ (related content set)}$$

Consider the content subsets $\mathcal{S}_k = \{k\} \cup \mathcal{R}_k$. Assume that only content k is stored in the cache ($x_k = 1$ and $x_n = 0, \forall n \neq k$). All requests for contents in S_k will be satisfied (i.e., "covered" by content k), and thus SCHR will be equal to $\sum_{i\in\mathcal{N}}\sum_{n\in S_k} p_n^i \cdot q_i$. When more than one contents are stored in the cache, let $\mathcal{S}^{'}$ denote the union of all contents covered by the stored ones, i.e., $\mathcal{S}^{'} = \bigcup_{\{k:x_k=1\}} S_k$. Then, the SCHR will be equal to $\sum_{i\in\mathcal{N}}\sum_{n\in S'} p_n^i \cdot q_i$. Hence, Problem (1) becomes equivalent to

$$\max_{\mathcal{S}^{'}} \sum_{n\in\mathcal{S}'} p_n^i \cdot q_i \qquad s.t. \quad |\{k : x_k = 1\}| \leq C.$$

This corresponds to the the *maximum coverage problem with weighted elements*, where "elements" (to be "covered") correspond to the contents $i \in \mathcal{K}$, weights correspond to the probability values $p_n^i \cdot q_i$, the number of selected subsets $\{k : x_k = 1\}$ must be less than C, and their union of covered elements is $\mathcal{S}^{'}$. This problem is known to be a NP-hard problem [25]. Consequently, the more generic problem with many helpers in range of each user (and potentially different u_{kn}^i and $0 \leq u_{kn} \leq 1$) is also NP-hard. □

Lemma 3 above states that finding the optimal placement is an NP-hard problem, *even if we had a single cache*. This is in contrast to the case without soft cache hits, where it is well known that the optimal policy for a single cache (or non-overlapping ones) is simple, namely to store the most popular

contents in every cache [5]. The following lemma however suggests that the problem is amenable to efficient approximate algorithms with provable performance guarantees.

Lemma 4. *Problem 1 has a submodular and monotone objective function (Eq.* (2)*), and a matroid constraint (Eq.* (3)*).*

Proof. The objective function of Eq. (2) $f(X) : \{0,1\}^{K\times M} \rightarrow \mathbb{R}$ is equivalent to a set function $f(S) : 2^{\mathcal{K}\times\mathcal{M}} \rightarrow \mathbb{R}$, where $\mathcal{K}\times\mathcal{M}$ is the finite *ground set* of tuples (k,j) ({content,helper}), and $S = \{k \in \mathcal{K}, j \in \mathcal{M} : x_{kj} = 1\}$. In other words,

$$f(S) \equiv \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot q_i \cdot \left(1 - \prod_{(n,j)\in S}\left(1 - u_{kn}^i \cdot q_{ij}\right)\right).$$

A set function is characterised as *submodular* if and only if for every $A \subseteq B \subset V$ and $\varepsilon \in V\backslash B$ it holds that

$$[f(A\cup\{\varepsilon\}) - f(A)] - [f(B\cup\{\varepsilon\}) - f(B)] \geq 0.$$

From Eq. (2), we first calculate

$$\begin{aligned} &f(A\cup\{(\ell,m)\}) - f(A) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \left(1 - \prod_{(n,j)\in A\cup\{(\ell,m)\}}\left(1 - u_{kn}^i \cdot q_{ij}\right)\right) \\ &\quad - \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \left(1 - \prod_{(n,j)\in A}\left(1 - u_{kn}^i \cdot q_{ij}\right)\right) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot u_{k\ell}^i \cdot q_{im} \cdot \prod_{(n,j)\in A}\left(1 - u_{kn}^i \cdot q_{ij}\right) \geq 0. \end{aligned}$$

Then,

$$\begin{aligned} &[f(A\cup\{(\ell,m)\}) - f(A)] - [f(B\cup\{(\ell,m)\}) - f(B)] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot u_{k\ell}^i \cdot q_{im} \cdot \prod_{(n,j)\in A}\left(1 - u_{kn}^i \cdot q_{ij}\right) \\ &\quad - \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot u_{k\ell}^i \cdot q_{im} \cdot \prod_{(n,j)\in B}\left(1 - u_{kn}^i \cdot q_{ij}\right) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \cdot u_{k\ell}^i \cdot q_{im} \cdot \prod_{(n,j)\in A}\left(1 - u_{kn}^i \cdot q_{ij}\right) \\ &\quad \cdot \left(1 - \prod_{(n,j)\in B\backslash A}\left(1 - u_{kn}^i \cdot q_{ij}\right)\right), \end{aligned}$$

The above is always ≥ 0, which proves submodularity. Finally,

$$\begin{aligned} f(B) - f(A) &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \left(1 - \prod_{(n,j)\in B}\left(1 - u_{kn}^i q_{ij}\right)\right) \\ &\quad - \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \left(1 - \prod_{(n,j)\in A}\left(1 - u_{kn}^i q_{ij}\right)\right) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} p_k^i \prod_{(n,j)\in A}\left(1 - u_{kn}^i \cdot q_{ij}\right) \\ &\quad \cdot \left(1 - \prod_{(n,j)\in B\backslash A}\left(1 - u_{kn}^i \cdot q_{ij}\right)\right), \end{aligned}$$

which is always non-negative, proving *monotonicity*.

To show that the constraint is a matroid (see, e.g., [26] for the definition of a matroid), we consider the set $\mathcal{V} = \mathcal{K}\times\mathcal{M}$ (i.e., all the possible tuples {content,helper}) and the collection of subsets of $\mathcal{V}$ that do not violate the capacity of the caches

$$I = \left\{S \subseteq 2^{\mathcal{V}} : |S \cap 2^{\{\mathcal{K},m\}}| \leq C, \quad \forall m \in \mathcal{M}\right\}.$$

Then: (a) For all sets A and B that $A \subseteq B \subseteq \mathcal{V}$, it holds that if $B \subseteq \mathcal{I}$ (i.e., the caching placement defined by B does not violate the size of the caches) then $A \subseteq \mathcal{I}$, because in A every cache has to store the same or less content than in B and thus no capacity constraint is violated.
(b) For all sets $A, B \in \mathcal{I}$ (i.e., feasible caching placements) and $|B| > |A|$ (i.e., in B more contents are cached), $\exists \ell \in B\backslash A$ that $A\cup\{\ell\} \in \mathcal{I}$, since in A not all caches are full (otherwise B would violate the capacity constraint, i.e., $B \notin \mathcal{I}$), which means that there exists at least one more content can be cached (and this content can be selected to be from the set B).

The matroid nature follows from (a) and (b) [26]. □

The above result suggests that a greedy algorithm can guarantee an $\frac{1}{2}$-approximation of the optimal solution [26], as we state in Theorem 5. Algorithm 1 is such a greedy algorithm, and is of computational complexity $O\left(K^2M^2\right)$.

Algorithm 1 Greedy Algorithm for Problem (1).
computation complexity: $O\left(K^2 \cdot M^2\right)$

Input: *utility* $\{u_{kn}^i\}$, *content demand* $\{p_k^i\}$, *mobility* $\{q_{ij}\}$, $\forall k, n \in \mathcal{K},\ i \in \mathcal{N},\ j \in \mathcal{M}$
$A \leftarrow \mathcal{K}\times\mathcal{M};\ S_0 \leftarrow \emptyset;\ t \leftarrow 0$
for $j \in \mathcal{M}$ **do**
  $c_j \leftarrow 0$
end for
while $A \neq \emptyset$ **do**
  $t \leftarrow t + 1$
  $(n, j) \leftarrow \underset{(k,\ell)\in A}{argmax}\ f(S_{t-1}\cup\{(k,\ell)\})$
  ▷ where, n: content; j: cache/SC
  if $c_j + 1 \leq C$ **then**
    $c_j \leftarrow c_j + 1$
    $S_t \leftarrow S_{t-1}\cup\{(n,j)\}$
  else
    $S_t \leftarrow S_{t-1}$
  end if
  $A \leftarrow A\backslash\{(n,j)\}$
end while
$S^* \leftarrow S_t$
return S^*

Theorem 5. *Let* OPT *be the optimal solution of Problem* (1)*, and* S^* *the output of Algorithm 1. Then, it holds that*

$$f(S^*) \geq \frac{1}{2} \cdot OPT.$$

Submodular optimization problems have received considerable attention recently, and a number of sophisticated approximation algorithms have been considered (see, e.g., [26] for a survey). For example, a better $\left(1 - \frac{1}{e}\right)$-approximation can be found following the "multilinear extension" approach [27]. Other methods also exist that can give an

$\left(1-\frac{1}{e}\right)$-approximation [28]. Nevertheless, minimizing algorithmic complexity or optimal approximation algorithms are beyond the scope of this paper. Our goal instead is to derive fast and efficient algorithms (like greedy) that can handle the large content catalogues and content related graphs $\mathbf{U}$, and compare the performance improvement offered by soft cache hits. The worst-case performance guarantees offered by these algorithms are added value.

IV. Performance Evaluation

A. Simulation setup

Content dataset. We consider a real dataset of YouTube videos from [29]. The dataset contains several information about the videos, such as their (a) popularity and (b) the list of related videos for each of them, as recommended by YouTube. This information can be used to choose p_k and u_{kn}. As the dataset does not contain per-user information, we consider the sub-case-2 of Definition 1 for our simulation evaluation.[4] After preprocessing the data to remove contents with no popularity values, we build the related content matrix (utility matrix $\mathbf{U}$). Due to the sparsity of the dataset, we only consider contents belonging to the largest connected component, that includes $K = 2098$ videos. The average number of related content for these videos is 3.6. For simplicity, we will initially assume that if content k is related to content n in the dataset, then $u_{kn} = 1$. However, we later perform a sensitivity analysis for diminishing acceptance probabilities, even for content recommended by the YouTube engine itself.

Cellular network. We consider an area of one square kilometre that contains $M = 20$ SCs. SCs are randomly placed (i.e., uniformly) in the area which is an assumption that has been also used in similar works [5], [30]. An SC can serve a request from a user when the user is inside its communication range, which we set to 200 metres. We also consider $N = 50$ mobile users. This creates a relatively dense network where a random user is connected to 3 SCs *on average*. We will also consider sparser and denser scenarios, for comparison. We generate a set of 20000 requests according to the content popularity calculated from the YouTube dataset, over which we average our results.

Unless otherwise stated, the simulations use the parameters summarized in Table II.

B. Performance Results

We consider the following four content caching schemes:

- *Single (popularity-based)*: Single cache accessible per user (e.g., the closest one). Only normal cache hits allowed, and the most popular contents are stored in each cache. This is the baseline scheme, commonly used in related works.
- *SingleSCH*: Single cache with soft cache hits, with the content allocation given by Algorithm 1.
- *Femto*: Femto-caching without soft cache hits (from [5]).
- *FemtoSCH*: Femto-caching with soft cache hits, with the content allocation given by Algorithm 1.

[4]We have performed simulations with per user utilities using synthetic data, and similar conclusions can be drawn.

TABLE II: Parameters used in the simulations.

Parameter	Value	Parameter	Value
nb. of contents, K	2098	nb. of requests	20000
Cache size, C	5	nb. of SCs, M	20
Area	1 km^2	Communication range	200 m

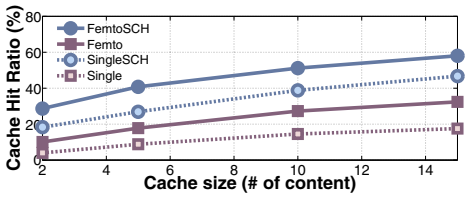


Fig. 2: Cache hit ratio vs. C, with fixed content size.

The single cache scenarios are considered because they are relevant in today's networks, where the cellular network first chooses a single base station to associate a user to (e.g., based on signal strength), and then the user makes its request [31]. This way we can compare how much of the caching gains comes from femto-caching (i.e. coverage overlaps) and which from SCHs.

Cache size impact: We first investigate the impact of cache size, assuming fixed content sizes. Fig. 2 depicts the total cache hit ratio, for different cache sizes C: we consider a cache size per SC between 2 and 15 contents. The simulations suggest that soft cache hits (SCH) can double the cache hit ratio. What is more, these gains are applicable to both the single cache and femto-caching scenarios, which show that our approach can offer considerable benefits *on top of femto-caching*. The two methods together offer a total of $3\times$ improvement compared to the baseline scenario "Single", reaching a maximum cache hit ratio of about 60% for $C = 15$. Finally, even with a cache size per SC of about 0.1% of the total catalogue, introducing soft cache hits offers 30% cache hit ratio, which is promising.

SC density impact: In Fig. 3 we perform a sensitivity analysis with respect to the number of SCs in the area (assuming fixed capacity $C = 5$). We test 2 sparse scenarios ($M = 5$ and $M = 10$) and 2 dense scenarios ($M = 20$ and $M = 30$). The average number of SCs that can be seen by a user varies from around 1 ($M = 5$) to 4.6 ($M = 30$). In the sparse scenarios "Femto" and "Single" perform similarly ($20 - 30\%$ cache hit rate) since a user can usually see at most one SC. As the SC density increases, the basic femto-caching is able to improve performance, as expected. However, femtocaching with SCH brings even more performance gains. With a storage capacity per SC of about 0.25% of the content catalogue (5 out of 2000 contents), and a coverage overlap of 2-4 SCs per user, femto-caching together with SCH can achieve a $30 - 50\%$ cache hit ratio. This is promising on the additive gains of the two methodologies.

Utility matrix impact: In these final two sets of simulations, we further investigate the impact of the content relations as captured by the matrix $\mathbf{U}$ (and its structure). A

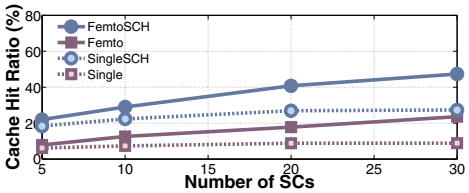


Fig. 3: Cache hit ratio vs. number of SCs M ($C = 5$).

first important parameter to consider is the average number of related contents per video, which we denote as $E[R]$, where the set R for a content k is defined as $R = \{n \in \mathcal{K} : u_{kn} > 0\}$. In the previous scenarios, the number of related contents was inferred from the YouTube trace, and was found to be equal to $E[R] = 3.6$. To understand the impact of this parameter, in this next scenario we generate two synthetic graphs $\mathbf{U}$:

- *SCH(1)*: content k is marked as related to content n with probability p_n (i.e., proportional to its popularity), normalized to a mean $E[R]$ value per content.
- *SCH(2)*: $E[R]$ related contents are selected randomly for each content.

While the latter assumes that content relations are independent of their popularity, the former assumes that a more popular content has a higher chance to appear in the related content list. In fact, this is quite inline with daily experience of how recommendation systems work.

In Fig. 4, we compare the cache hit ratio for single and femto-caching scenarios: without SCH, with SCH(1), and with SCH(2), assuming that $\mathbf{E}[R]$ varies between 2 and 10 related content items. A first observation is that, due to the sparsity of the content matrix, SCH(2) (i.e., random content relations) brings only marginal improvements to the total number of hits. On the other hand, a correlation between related content and popularity (i.e., SCH(1)) is what brings considerable offloading gains, even for small $E[R]$. In fact, comparing these synthetic results with the previous trace-based ones, one can infer that the real dataset probably more closely resembles SCH(1), i.e., does exhibit such a correlation.

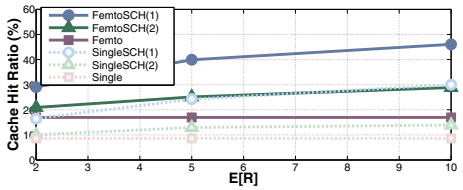


Fig. 4: Cache hit ratio for different number of related contents $E[R]$; synthetic traces.

User flexibility: In this last scenario, we present in Fig. 5 the cache hit ratio as a function of the willingness of a user to accept a related content. We consider two scenarios, both in the femto-caching context:

- *Synthetic:* We generate a synthetic matrix U as in SCH(1) above, with $E[R] = 4$ and $u_{kn} = u < 1$.
- *YouTube:* We use the real YouTube dataset for the matrix U. However, all related contents also have a utility $u_{kn} = u < 1$ (instead of $u_{kn} = 1$ considered in the previous scenarios).

On the x-axis of Fig. 5 we vary this parameter u from 0 to 1. As can be seen there, when the user's acceptance probability becomes very small, the scenario becomes equivalent to standard femto-caching without soft cache hits, and the gains reported there are in line with the previous plots. However, as user willingness to accept related content increases, the optimization policy can exploit opportunities for potential soft cache hits and improve performance. E.g., for a probability 50% to accept an alternative recommended content, cache hit ratios increase by almost $2\times$ (from 15% to 27% in the YouTube dataset). Results are in fact very comparable for the synthetic and YouTube traces.

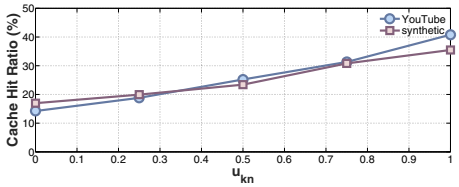


Fig. 5: Cache hit ratio as a function of user's willingness to accept related content ($M = 20, C = 5, E[R] = 4$).

V. Related Work

Mobile Edge Caching. Densification of cellular networks, overlaying the standard macro-cell network with a large number of SCs (e.g., pico- or femto-cells), has been extensively studied and is considered a promising solution to cope with data demand [32], [33], [34]. As this densification puts a tremendous pressure on the backhaul network, researchers have suggested storing popular content at the "edge", e.g., at SCs [5], user devices [8], [9], [12], or vehicles [10], [11].

Our work is complementary to these approaches, as it can utilize such mobile edge caching systems while showing how to further optimize the cache allocation when there is a cache-aware recommender systems in place. We have applied this approach in the context of mobile (ad-hoc) networks with delayed content delivery [21] as well, and applied it here for the first time in the context of femto-caching [5]. Additional research directions have also recently emerged, more closely considering the interplay between caching and the physical layer such as Coded Caching [7] and caching for coordinated (CoMP) transmission [35], [36]. We believe the idea of soft cache hits could be applied in these settings as well, and we plan to explore this as future work.

Caching and Recommendation Interplay. There exist some recent works that have jointly considered caching and recommendation for wireless systems [14], peer-to-peer networks [37], and CDNs [38], [13]. Specifically, [37] studies the interplay between a recommendation system and the performance of content distribution on a peer-to-peer network like BitTorrent (e.g., recommending contents based on the number of "seeders") towards improving performance.

[38] shows that users tend to follow YouTube's suggestions, and despite the large catalog of YouTube, the top-10 recommendations are usually common for different users in the same geographical region. Hence, CDNs can use the knowledge from the recommendation system to improve their content delivery. Finally, [13], [14] propose approaches of recommended list reordering, which can achieve higher cache hit ratios (e.g., for YouTube servers/caches).

These works, except for [14] which is closer to our study, (i) focus on the recommendation side of the problem, ignoring or simplifying the optimal caching algorithm, and (ii) do not consider the wireless cooperative caching aspect of the problem. Nevertheless, the increasing dependence of user requests on the output of recommender systems clearly suggests that there is an opportunity to further improve the performance of (mobile) edge caching by jointly optimizing both, with minimum impact on user Quality of Experience.

VI. Conclusions

In this paper, we have proposed the idea of *soft cache hits*, where an alternative content can be recommended to a user, when the one she requested is not available in the local cache. While normal caching systems would declare a cache miss in that case, we argue that an appropriate recommended content, related to the original one can still satisfy the user with high enough probability. We then used this idea to design such a system around femto-caching, and demonstrated that considerable additional gains, *on top of those of femto-caching* can be achieved using realistic scenarios and data. We believe this concept of soft cache hits has wider applicability in various caching systems.

References

[1] 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.

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

[3] G. S. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16–22, 2016.

[4] G. Paschos and P. Elia, "Caching for wireless networks," in *IEEE Sigmetrics Tutorials*, 2016.

[5] 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.

[6] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3665–3677, 2014.

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

[8] P. Sermpezis and T. Spyropoulos, "Effects of content popularity on the performance of content-centric opportunistic networking: An analytical approach and applications," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3354–3368, 2016.

[9] 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*, 2012.

[10] 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.

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

[12] P. Sermpezis and T. Spyropoulos, "Offloading on the edge: Performance and cost analysis of local data storage and offloading in HetNets," in *Proc. IEEE WONS*, pp. 49–56, 2017.

[13] D. K. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-centric video recommendation: an approach to improve the efficiency of youtube caches," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 11, no. 4, p. 48, 2015.

[14] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *Proc. IEEE INFOCOM*, 2017.

[15] "T-Mobile Music Freedom." https://www.t-mobile.com/offer/free-music-streaming.html, 2017.

[16] Y. Yiakoumis, S. Katti, and N. McKeown, "Neutral net neutrality," in *Proc. ACM SIGCOMM*, pp. 483–496, 2016.

[17] S. Guo, M. Derakhshani, M. Falaki, U. Ismail, R. Luk, E. Oliver, S. U. Rahman, A. Seth, M. Zaharia, and S. Keshav, "Design and implementation of the kiosknet system," *Computer Networks*, vol. 55, no. 1, pp. 264–281, 2011.

[18] "Internet.org by Facebook." https://info.internet.org/, 2017.

[19] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing a key technology towards 5g," *ETSI White Paper No. 11*, 2016.

[20] C. Liang and F. R. Yu, "Wireless network virtualization: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358–380, 2015.

[21] T. Spyropoulos and P. Sermpezis, "Soft cache hits and the impact of alternative content recommendations on mobile edge caching," in *Proc. ACM Workshop on Challenged Networks (CHANTS)*, pp. 51–56, 2016.

[22] P. Sermpezis, T. Spyropoulos, L. Vigneri, and T. Giannakas, "Femto-caching with soft cache hits: Improving performance through recommendation and delivery of related content," *available at https://arxiv.org/abs/1702.04943*, 2017.

[23] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, pp. 4:2–4:2, 2009.

[24] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[25] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[26] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, no. 19, p. 8, 2012.

[27] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a monotone submodular function subject to a matroid constraint," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.

[28] Y. Filmus and J. Ward, "Monotone submodular maximization over a matroid via non-oblivious local search," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 514–542, 2014.

[29] http://netsg.cs.sfu.ca/youtubedata/, 2007.

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

[31] S. Sesia, I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009.

[32] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, J. G. Andrews, P. Xia, H. S. Jo, H. S. Dhillon, and T. D. Novlan, "Heterogeneous cellular networks: From theory to practice," *IEEE Comm. Magazine*, vol. 50, no. 6, pp. 54–64, 2012.

[33] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M. C. Reed, "Femtocells: Past, present, and future," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 3, pp. 497–508, 2012.

[34] J. G. Andrews, "Seven ways that hetnets are a cellular paradigm shift," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 136–144, 2013.

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

[36] A. Liu and V. K. Lau, "Cache-enabled opportunistic cooperative mimo for video streaming in wireless systems," *IEEE Trans. Signal Processing*, 2015.

[37] D. Munaro, C. Delgado, and D. S. Menasché, "Content recommendation and service costs in swarming systems," in *Proc. IEEE ICC*, 2015.

[38] D. K. Krishnappa, M. Zink, and C. Griwodz, "What should you cache?: a global analysis on youtube related video caching," in *Procc. ACM NOSSDAV Workshop*, pp. 31–36, 2013.