Computer Networks 110 (2016) 306–323

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Addressing niche demand based on joint mobility prediction and content popularity caching

Xenofon Vasilakos [a,*], Vasilios A. Siris [a], George C. Polyzos [a]

Mobile Multimedia Laboratory, Department of Informatics School of Information Sciences and Technology, Athens University of Economics and Business 76 Patision Str., 104 34, Athens, Greece

ARTICLE INFO

Article history:
Received 13 January 2016
Revised 31 July 2016
Accepted 2 October 2016
Available online 8 October 2016

Keywords:
Niche demand
Proactive caching
Mobile networks
Mobile video
Content popularity

ABSTRACT

We present an efficient mobility-based proactive caching model for addressing *niche* mobile demand, along with popularity-based and legacy caching model extensions. Opposite to other proactive solutions which focus on popular content, we propose a distributed solution that targets less popular, personalised or dynamic content requests by prefetching data in small cells based on aggregated user mobility prediction information. According to notable studies, niche demand, particularly for video content, represents a *significant* 20–40% of Internet demand and follows a growing trend. Due to its novel design, our model can directly address such demand, while also make a joint use of content popularity information with the novelty of dynamically *tuning* the contribution of mobility prediction and content popularity on local cache actions.

Based on thorough performance evaluation simulations after exploring different demand levels, video catalogues and mobility scenarios including human walking and automobile mobility, we show that gains from mobility prediction can be high and able to adapt well to temporal locality due to the short timescale of measurements, exceeding cache gains from popularity-only caching up to 41% for low caching demand scenarios. Our model's performance can be further improved at the cost of an added computational overhead by adapting cache replacements by, e.g. in the aforementioned scenarios, 41%. Also, we find that it is easier to benefit from requests popularity with low mobile caching demand and that mobility-based gains grow with popularity skewness, approaching close to the high and robust gains yielded with the model extensions.

1. Introduction

Wireless networking witnesses an unparalleled growth. During the past ten years mobile data demand grew x4000 and only in 2015 by 74% [1]. According to the same study, mobile data per device will grow to 4.4 GB per month by 2020, which is almost a x5 increase relative to 2015. *Mobile video* on its own will account for 75% of traffic, pushing macro-cellular capacity to the limit. In another report in [2], wireless communication systems will support more than 1000× today's traffic beyond 2020.

This unprecedented growth calls for cost-efficient solutions that do *not* relegate user Quality-of-Experience QoE. To this end, several solutions in literature try to offload [3] the costly macro-cellular traffic to *small cells* as an alternative to expanding macro-cellular coverage to satisfy demand. Small cells can be provider-controlled *pico*/*femto* cells or third-party open-access Wi-Fi hot-spots. Due to their small range, they allow providers to better *utilise* the licensed spectrum, to exploit the *non*-licensed spectrum through of Wi-Fi hotspots, and to reduce power consumption for signal emissions with significant benefits for providers' costs and user's battery consumption. On the down side, installing provider-controlled small cells in places that lack or have poor backhaul support, e.g., on lamp posts or traffic lights, implies considerable business expenditures for deploying, running and maintaining new infrastructures and services that can meet with demand volumes. In addition, exploiting the already established Wi-Fi hotspot infrastructure is also limited due to the capacity restrictions of typical backhaul connections that create a *bottleneck* for certain throughput-demanding content types and, particularly, for mobile *video*: Even if higher capacities are available, users tend [1] to cause even larger traffic volumes when offered with higher capacities.

To address the problem of backhaul congestion, recent developments in the literature [4–8] as well as in the industry[1] adapt *proactive* caching of content in small cells so as to directly serve

* Corresponding author. Tel.: +30 210 8203 693.
E-mail addresses: xvas@aueb.gr (X. Vasilakos), vsiris@aueb.gr (V.A. Siris), polyzos@aueb.gr (G.C. Polyzos).

[1] For instance, Altobridge "Data at the edge" www.idirect.net/Altobridge.aspx or www.intel.com/content/dam/www/public/us/en/documents/white-papers/communications-small-cell-study.pdf.

http://dx.doi.org/10.1016/j.comnet.2016.10.001
1389-1286/

mobiles from a low-access delay local cache. Similarly to Content Delivery Network (CDNs), these local approaches focus on serving *popular* content to users, hence leaving *niche* demand for less popular or personalised content that falls in the "long tail" of typical popularity distributions largely unaddressed.

In this paper we take a different approach and propose a novel, mobility- and popularity-based distributed proactive caching model with notable design differences from the other solutions, which *focuses on niche* mobile video demand. This way, local cache actions target less popular content or personalised requests which are *not* directly addressed by CDNs, thus our model's performance can be more robust w.r.t. the already established and successful CDN replication of popular content near to consumption, which leaves little room for further improving the service and related user QoE of popular mobile videos. Cache decisions with our model are lightweight and taken *autonomously* by each small cell based on a dynamic cache congestion pricing scheme that helps to efficiently trade the limited local cache buffer resources for reduced user download charges or experienced delay. Therefore, our solution can be easily applied to heterogeneous wireless networking scenarios to yield gains with a positive implication on QoE for monetary cost-concerned or delay-sensitive mobile users, via trading pure macro-cellular communication with small-cellular downloads.

1.1. Contribution

The contribution of this paper can be summarised as:

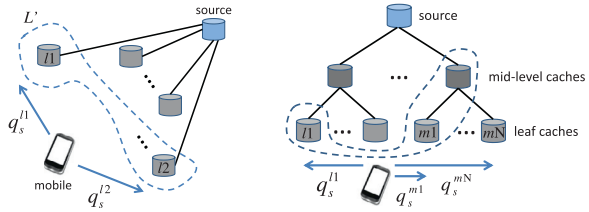
• *Addressing niche demand with mobility-based proactive caching:* We conclude after notable studies [9–13] that a significant 20–40% of requests that refers to less popular videos remains not addressed by popularity-based approaches. Thus,

> *even if 60–80% of video accesses continuous to account for popular content in the future, a significant 20–40% of the total mobile video requests will still refer to niche demand.*

Moreover, we discuss the possible implications of niche personalised demand in social networks and try to explore the possible impact of CDNs on the delay gains from local caching, showing that our model can have a good and robust performance against the cases of applying no, popularity-only or naïve local proactive caching, exactly due to targeting the part of niche mobile demand which is not directly addressed by CDNs.

• *Joint mobility-based and popularity caching:* We present Efficient Mobility-based Caching (EMC), a proactive caching model that is designed to address *niche* demand for personalisedor less popular requests that fall within the "long tail" of typical popularity distributions via cache decisions which exploit individual user *mobility* prediction and requests. We also present content *popularity* and legacy-caching model extensions that aim to serve multiple mobiles sharing the same requests. The main *novelty* of the presented approach lies in the abilityto dynamically *tune* the contribution of contemporary dynamic mobility and popularity information on cache actions and, hence, to capture short timescale temporal locality better than other approaches.

• *Investigation of influence of system parameters on performance:* We investigate the impact of mobility-prediction and popularity on the gains from proactive caching for a plethora of different system parameters. We find that gains from mobility prediction can be high and able to adapt well to temporal locality due to the short timescale of measurements, and that it is easier to benefit from requests' popularity under low mobility-based caching demand conditions. Furthermore, the gains of our basic mobility-based model grow with video popularity skewness, approaching close to the high and robust gains yielded with the model extensions.



(a) A "flat" neighbourhood of caches. (b) A two-level cache hierarchy.

Fig. 1. Independent prefetching decisions in (a). Cooperation between mid-level and leaf caches in (b).

1.2. Outline

This paper is organised as follows: Section 2 provides the necessary background on the problem of mobile video explosion and a thorough discussion on the significance of niche mobile video demand. We proceed with presenting the basic system model along with popularity and legacy caching extensions in Section 3. Next, we analyse the results of a comprehensive performance evaluation in Section 4, followed by an analytical discussion on implementation in Section 5. Finally, we discuss the state-of-the art in Section 6 before we wrap up and outline future work directions in Section 7.

2. Background and motivation

Mobile video traffic arises as the *ultimate* challenge for wireless providers. Due to the combined widespread of devices and video content, particularly within social networks, mobile video is expected to account for 75% of traffic by 2020 [1]. This translates to a x11 demand increase relative to only 2015, and is impossible or too expensive to satisfy by simply expanding the traditional macro-cellular support. The problem gets further aggravated due to the aggressive buffering policies of most HTTP video streaming services which cause 25–39% of unnecessary mobile traffic [14]. Another important problem dimension regards service charges. Users can become very unhappy with charging mechanisms that aim to control demand peaks.

Perhaps the most difficult dimension of the problem which we try to address in this paper regards capturing *niche* demand. The currently applied solutions in the internet help to mitigate the problem of increased demand by identifying which content is popular and by increasing its availability. CDNs, in particular, are highly successful in replicating popular content close to consumption, which helps to reduce providers' cost and to offer better services to their users. But as we discuss, even CDNs fail to capture big part of mobile demand which is due to less popular content. This part of traffic has an increasing significance due to the increasing share of video content in *social networks*, which create a trend for sharing personalised, hence less popular, videos.

In the rest of this section we focus on the background details of the problem, explain why niche demand is significant and argue why it will continue to be at least as important in the future due to social networking. Moreover, we discuss why we need locally applied solutions in small cells targeting on niche rather than popular videos, which can complement CDNs in capturing the part of the traffic that they typically neglect, yielding benefits for both users and providers.

2.1. Niche demand is significant

Regarding *web content*, Fig. 1 in [9] shows that popularity-based caching reaches a 60–70% practical *hit ratio upper limit* depending on the homogeneity of the client population. A similar conclusion

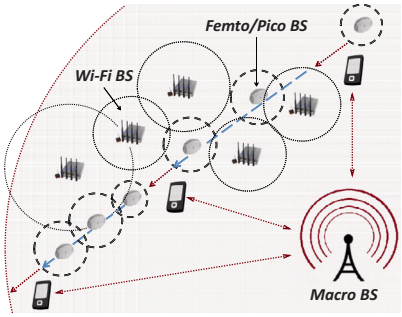


Fig. 2. Macro-cellular area including provider's pico/femto cells and third-party Wi-Fi hotspots. Blue-dashed (resp. red-dotted) arrows denote that the mobile is connected to a small (resp. the macro) BS.

is drawn from Fig. 2 in [10] which shows that 25–40% of the top-most popular documents account for 70% of all requests seen by a web cache proxy. In a more recent study in [15], the authors shows that a large portion of web traffic is *dynamically* generated and *personalized* data, contributing up to 30–40% of the total traffic. These findings lead to the conclusion that *at least a significant 30% of less popular or personalised web requests that account for 30–40% of traffic, are not addressed based on popularity.*

Regarding *video demand*, the work in [12] and references therein, acknowledge the significance of niche demand and identify the opportunities of leveraging *latent* demand for niche videos that are currently not reached. According to the same study, caching only 10% of the most popular videos accounts for 80% of views on YouTube, i.e. a 20% of videos accesses on YouTube corresponds to less popular content. However, this profound popularity skewness could be the result of a *distortion* imposed by search or recommendation algorithms due to wrong categorisation or ranking. Speculation is also backed by another study [13] on different centralised video services, which reveals that 10% of the top-most popular videos accounts for ~60% of requests, i.e. 90% of the *least* popular videos accounts for *40% of all requests*. For all that, we can conclude that a *significant* part of video requests, which reportedly lies between 20% and 40%, is *not* addressed by popularity-based caching, hence *even if 60–80% of video accesses continuous to account for popular content in future, a significant 20–40% of the total mobile video demand will still refer to niche content.*

2.2. Video demand in Social Networking Service (SNS)

Social Networking Service (SNSs) reflect social relations among users who share a common background such as news interests or real-life connections, thus forming a unique *personalized* online social network per user profile. Since their advent, they are having an increasing impact on ISP traffic and on wireless mobility: Cisco [1] verifies that the observed increase in wireless traffic volumes is the result of the increasing popularity of SNSs and predominantly Facebook, in addition to the traffic from large video streaming providers like YouTube or Netflix. While initially focused mostly on textual and image content, currently SNSs mark *a new era of personalised videos*. Before that, demand was driven by developments in large video streaming providers, which led to solutions designed for increasing the offered QoS of such services. In what follows, we focus on the current and future levels of mobile video traffic on Facebook and stress the *qualitative* differences between SNSs and video hosting websites.

2.2.1. Video explosion on Facebook

Facebook is the pacesetter for almost all developments in SNSs, marking the trend for new mobile services. As of 2010 [16], its users where uploading over a billion of new images, i.e. 60 TB, per week, with Facebook serving over *one million images per second* during peak hours. But clearly Facebook content sharing has shifted from image data to an era of *video dominance*. Video functionality was added on Facebook in 2007, and since then video posts have gradually evolved from simple external links to uploaded videos on Facebook facilities[2] Moreover, the latest web and mobile versions of Facebook enable video *auto-play* on users "timelines" or profile pages. This is important for it implies an added traffic for videos that are *not* explicitly requested to be viewed by users, similarly to aggressive buffering techniques [14] applied by services like YouTube. Finally, files will be *significantly bigger* in *future*, as advanced mobile devices and greater network capacities will enable users to capture and upload videos with a longer duration and in a higher resolution.

2.2.2. Differences from video hosting websites

To complete one's understanding on the level of impact of SNSs on video demand, we must focus on the differences w.r.t. the demand models in the case of "traditional" video streaming services such as YouTube. The latter holds a leading role [17] and, consequently, it is a de facto case-study [12,14] for Internet video demand. Prospective viewers are offered with a keyword-based *search* mechanism, paralleled by *suggestions* based on popularity trendsin order to *discover desired* videos. As identified in [12] and references therein, the *filtering* mechanisms applied on search results and on recommended content can be held responsible for the occurring *truncated power-law distributions*, i.e. distributions characterised by only *a small fraction* of videos that *stands out substantially* for its *popularity* and, therefore, translates to most of the produced traffic volumes. With this in mind, CDN services (see Section 2.3) and local caching strategies (see Section 6.1) are designed to bring popular content "closer" to its prospective viewers. Nonetheless, this approach may be not appropriate w.r.t. SNS demand, due to the following important *qualitative differences* from the demand model in the case of "traditional" video streaming services:

2.2.2.1. User-specific filtering. Video hosting websites try to influence popularities [12]. Unlike that, demand in SNSs is mainly dictated by the users' personalized networks of peers, liked/followed accounts and preference settings. For instance, Facebook users do *not* directly search for content based on keywords. Their "timeline" of content suggestions is defined after their own settings used as input to a special algorithm which features only desired content from *within their social network* instead. Twitter on the other combines personalised (filter) listswith hashtag searching and news "trends".

2.2.2.2. Individuality, transitiveness & transience. Popularity depends on the characteristics of the *individual* users' social networks. To provide an example, the chance that a published video becomes popular grows with the publisher's network size. Also, it is highly *transitive* as content can gain increased visibility if republished by popular users. Nevertheless, popularity falls sharply after two days on Facebook because publications stop from showing in the default news feed [16].

2.2.2.3. Latent popularity. Latent popularity is difficult to capture. Contacts may consume content without commenting, liking, etc. Even views-counting provides a questionable measure of popularity –especially now that users can "silently" watch or neglect autoplayed videos – with further studies [14] showing that users can

[2] According Facebook's CEO M. Zuckerberg, "*In five years, most of [Facebook] will be video*" http://www.pcworld.com/article/2844852/facebook-will-be-mostly-video-in-5-years-zuckerberg-says.html.

easily abort playback. And although there appears to be a strong correlation between popularities and rankings in [12], the authors of the same study also state that requests on YouTube are so skewed that it is *debatable* whether this is because viewers want to watch what others watch or because of a distortion caused by wrong ranking or categorisation.

2.3. The impact of CDNs on local caching

Content Delivery (or Distribution) Networks (CDNs) are large distributed systems deployed within ISP facilities across the Internet with the goal to serve content with high availability and performance. To do so, they exploit the truncated power-law popularity distributions of web and video demand and use distributed algorithms [18] to bring *popular* objects *"closer"* to prospective *end-users*. This way, CDNs also reduce data transit charges [19], avoid bandwidth bottlenecks and large network distances.

Still, CDNs can fail to assist a considerable part of demand: Cache *effectiveness* for content in the "long tail" of popularity distributions increases *only logarithmically* with the size of the cache [10,20], whereas CDN edge-caches are sufficiently big for hosting *only* the most popular videos. This leaves *niche* requests to "pass through" the CDN level, although they represent a significant portion of mobile demand (see Sections 2.1 and 2.2). Moreover, there is evidence that content popularity is *not* adequate for intercepting post-CDN demand: Empirical findings [16] indicate that requests missed by the CDN caches are *unlikely* to hit Facebook's internal cache due to wrong cache actions as a result of a plethora of content of similar popularity in the "long tail". Likewise to Facebook's internal cache misses, local caches deployed on networks edges that adapt a popularity-based caching model can also leave niche demand unaddressed.

For all that, we argue for a local solution, that is beyond the scope of popularity-based approaches, and which can *complement* CDNs by addressing niche mobile video requests. To our knowledge, EMC is currently the only such solution in literature.

3. System model

We present our basic mobility-based cache decisions model and how it can be applied in both a "flat" and a hierarchical configuration of the available distributed cache space (Section 3.1). Furthermore, we present content popularity and legacy caching model extensions (Section 3.2), and demonstrate an application scenario for users of heterogeneous wireless networks who wish to reduce their monetary charges (Section 3.3). Last, we discuss and an analyse how (i) monetary charges or (ii) delay costs and their impact on QoE can be integrated in our model decisions (Section 3.4).

3.1. Basic model

Cache actions with the basic Efficient Mobility-based Caching (EMC) model consider (i) the cost of fetching data from an *expensive* source, e.g. from a mobile carrier's macro cell or a remote source T_R, and (ii) the cost for consuming data from the local *cache point* T_L.

"Cache points" are mobile network Access (or Attachment) Point APs equipped with low-access time storage resources which take cache actions that aim to *trade* their locally available buffer space efficiently for a reduced cost of obtaining data by their ephemerally hosted mobiles. Our model can be applied when $T_L < T_R$, i.e. when caching content closer to consumers or within the provider' own resources reduces transfer delay or data consumption charges for mobiles. Both T_R and T_L, which we discuss in more detail in Section 3.4 on page 17, can be assessed per individual content or even content chunks, or per user. Additionally, such costs can be associated with certain content types (e.g. premium quality videos) and/or Quality-of- Service (QoS) requirements for certain categories of mobiles and applications.

In addition, cache actions consider the cost of consuming cache space in order to capture the *trade-off* between occupying cache space for some content instead of another. Since EMC offers a caching service to both niche and popular requests, demand expectations can be higher compared to popularity-based solutions. In order to tackle cache congestion and to utilise the local storage resources efficiently, EMC uses a *dynamic price* $p_l(t)$ that is updated with each new request s issued at time $t+1$ to cache point l with capacity B_l:

$$p_l(t+1) = \left[p_l(t) + \gamma \sum \left(o_s \cdot b_s(t) - B_l\right)\right]^+. \tag{1}$$

Parameter γ defines how quickly prices adapt to demand changes, o_s is the size of each content s, and b_s is a binary value which denotes if s will be cached. Adjusting the price helps to efficiently utilise the cache and to reach closer to the cost *optimisation target*:

$$\min_{b_s} \quad \sum_{s \in S_l} \mathcal{T}_s \tag{2}$$

$$\text{s.t:} \quad \sum_{s \in S_l} o_s \cdot b_s \leq B_l\,, \quad \forall l \in L, \tag{3}$$

where $\mathcal{T}_f$ stands for the expected cost for obtaining content s. Note that trading transfer and cache costs presupposes a common measurement *value*, such as a monetary unit, which expresses how much willing are mobile applications to "pay" in terms of cache storage in exchange for reduced transfer costs for obtaining s.

3.1.1. Autonomous caching in a "flat" structure of cache points

Fig. 1 a depicts a neighbourhood of caches organised in a "flat" structure in which cache decisions $b_s^l(t)$ in every cache point l are *autonomous* according to rule (4):

$$b_s(t) = \begin{cases} 1 & \text{if } Q_s^l(T_R - T_L) \geq p_l(t), \\ 0 & \text{Otherwise}\,. \end{cases} \tag{4}$$

The rule captures user mobility prediction and individual requests information via Q_s^l, which aggregates the individual transition probabilities $q_s^{i,l}$ of *all* mobiles with an active request for object s from their current cache point i to the target point l:

$$Q_s^l = \sum\nolimits_{\forall i} q_s^{i,l} \tag{5}$$

The left-hand side the rule is divided by the object's size to adapt the rule for different-sized objects, i.e. $Q_s^l \cdot (T_R - T_L)/o_s$. Also, the rule can be accordingly adjusted to address cases in which transfer costs are individual to different content, caches, users, etc, hence the latter can be accordingly adapted as T_L^s and T_R^S in (4).

Following a mobile's handover to l, (i) the mobile starts to consume s from the local cache (in case of a cache hit), while (ii) the rest of caches l' in the neighbourhood of i get notified to reevaluate $Q^{l'}$ and accordingly to redecide upon evicting or keeping s. Likewise, l redecides upon keeping s or not after its data are transferred to the mobile.

Regarding *where* decisions are taken, one option is for the requesting mobile (or some proxy) to inform the neighbouring caches about the corresponding mobile's transition probabilities; alternatively, the mobile (or its proxy) can decide after learning the transfer costs and the cache prices from all the neighbours. As for *when* should prefetching start and for *which parts* of the content, this is a function of (i) the mobile's expected handover and connection (a.k.a. residence) durations to the next cache and (ii) the time needed for prefetching data from the remote location(s) to the cache.

3.1.2. Cooperative caching: a two-level hierarchy

EMC can be also applied in a *hierarchical* cache space where *leaf* caches are under only one *mid-level* cache and the cost for fetching data from the mid-level parent T_M satisfies $T_L < T_M < T_R$. For practical reasons our analysis is specifically focused on the case of a *two-level* cache hierarchy (see Fig. 1b), because this configuration reflects the current network reality where leafs correspond to local networks such as small cells or wired office LANs, and mid-level caches to CDN servers placed inside Internet Service Providers (ISPs). Generalising to arbitrary hierarchies is possible via a recursive application of the decision procedure which we discuss next, yet this comes at a cost of added complexity with limited practical relevance.

Our approach for solving the proactive caching problem in a two-level hierarchy considers two flat cache selection problems: One assuming that content is proactively fetched in the mid-level, and the other assuming that content is *not* proactively fetched in the mid-level cache. Each of these problems can be solved using the distributed approach presented in Section 3.1.1 by having the mid-level to submit the values of T_R for obtaining data from its original source to the leafs, and T_M for obtaining data from the mid-level cache. Next, each leaf uses (4) to decide for each of the two problems: (i) for the problem where the content is assumed to be *not* cached its parent, the leaf uses formula (4) to decide if the object should be prefetched to the leaf, and (ii) for the problem where the content is assumed to be cached in its parent, the leaf uses (4) by replacing T_R with T_M to decide. Then, each leaf informs its mid-level parent about the corresponding resulting average data transfer cost: $q_s^{i,l}T_R$ or $q_s^{i,l}T_M$ if the decision is to *not* fetch the object to the leaf, or otherwise $q_s^{i,l}T_L$.

After receiving the costs from all its leafs, each mid-level cache considers the sum of the transfer costs for each problem: D_M^{mid} in the case the mid-level proactively fetches the content and D_{bkhl}^{mid} in the case the mid-level cache does not prefetch the content but, instead, it will bring it from its backhaul upon request. The verdict $b_s^{mid}(t)$ upon caching an object in the mid-level resembles rule (4):

$$b_s^{mid}(t) = \begin{cases} 1 & \text{if } \mathcal{D}_{bkhl}^{mid} - \mathcal{D}_{M}^{mid} \geq p_{mid}(t), \\ 0 & \text{Otherwise}, \end{cases} \tag{6}$$

where $p_{mid}(t)$ is the congestion price for the mid-level at the time of the decision t, updated in a similar manner as for the leaf caches (1), only based on the cache demand and the available storage in the mid-level. Following its decision, the mid-level informs the leafs which transfer cost factor (T_R or T_M) they should use in (4), implying that the content may be proactively fetched even to both the leaf and its mid-level parent.

The above procedure remains *distributed* as in the case of the flat structure, but requires *cooperation* between mid-level caches and their leafs. Moreover, it can be applied to a hierarchy with more than one mid-level caches, as long as each leaf is a child of only one mid-level cache. Cace actions for each mid-level cache and its leafs follows the above approach, independently from the other mid-level caches.

3.1.3. Problem hardness and the feasibility of an optimal solution

3.1.3.1. Flat cache configuration. Finding an optimal cache placement for a flat set of caches given a set of requests for *equally-sized* objects of content can be obtained as follows: For each cache l we order requests in a decreasing order of value $q_s^{i,l}(T_R - T_L)$. Then, starting from the request with the highest $q_s^{i,l}(T_R - T_L)$, we fill the cache until the constraint B_l is reached. This procedure for obtaining the optimal is performed in *rounds*, unlike the *online*-applied solution based on cache congestion pricing, where cache actions are taken iteratively for each request according to formula (4). Furthermore, there is a serious practical issue regarding the duration of each round, which determines the number of requests that are considered in the beginning of a round. For objects with *different sizes*, the optimisation problem becomes identical to the *0/1 Knapsack problem*, which falls within the class of NP-hard problems and for which the cache congestion pricing can have advantages towards approaching an optimal solution.

3.1.3.2. Two-level cache hierarchy. Introducing an intermediate level of caching between the leaf caches and the remote data sources turns the problem into a Generalised Assignment problem [21]. The latter is a variation of the 0/1 Multiple Knapsack problem, involving multiple knapsacks, each with a possibly different maximum weight limit. At any time, there will be a given set of cache requests from the mobiles that are active at that time instant. For each such time instance, the problem for a two-level cache hierarchy has similarities with the *Data Placement Problem* [22], where the probability of an object being requested at a specific cache is given by the probability of the mobile moving to the corresponding network attachment point. In another work by the same authors [23], it is shown that the data placement problem with different object sizes is NP-complete. Although the problem may be solved in polynomial time [24] in a hierarchical network with equally-sized objects, such solutions have a *high* polynomial degree and apply to an offline version of the problem.

3.2. Model extensions

We extend our basic model to integrate cache *replacements* and/or to jointly exploit content *popularity* information along with user mobility prediction for cache actions. This yields two alternative extended model versions:

1. Efficient Mobility-based Caching with Replacements (EMC-R), and
2. Efficient Mobility and Popularity-based Caching with Replacements (EMPC-R)

The goal of the adapted extensions is twofold: it lies in better capturing temporal locality so as to prefetch and keep content that is more likely to be served to multiple users, and second, in exploiting *legacy-cached* content from past decisions in the case of EMPC-R, for which there are currently no active mobile requests.

3.2.1. Cache replacements

Adapting cache replacements causes two important differences relative to the basic model: (i) requests are not directly tested with a decision rule like (4) and (ii) actions do *not* consider local cache prices. Instead, the extended model directly caches an object s if its size o_s can fit in the unallocated cache space, otherwise it explores the possibility of evicting one or more cached objects e, either legacy or active. To decide which object(s) to evict, we follow a procedure according to which cached objects e with size o_e are polled for eviction in order of increasing $G(e)/o_e$ untilthere is enough

$$\sum\nolimits_{\forall e} G(e) / \sum o_e < G(s)/o_s \tag{7}$$

free space cache s under *constaint* (7), where $G(x)$ is the expected gain from caching (resp. keeping cached) content x, computed after a special *gain valuation formula* that is subject to the considered requests information (see Section 3.2.2). If (7) is not satisfied, then the cache request for s gets dismissed. The purpose of (7) is to optimise the *total gain per utilised cache buffer unit* and it can be omitted if all objects have the same size, e.g. when cache decisions are taken on the level of equally-sized content chunks. Note that (7) works also as a heuristic for tackling the knapsack combinatorial optimisation problem that arises from maximising the total gain of the cached objects given their different individual gains, sizes, and the limited capacity of the cache.

3.2.2. Extended model versions

Next, we present two alternative extended model variations w.r.t. gain valuation. Notice that only EMPC-R can exploit the possible benefits of legacy cached contents e for which there are no active mobile requests, i.e. for which $Q_e^l = 0$.

1) *Efficient Mobility-based Caching with Replacements (EMC-R).* This variation adapts only the cache replacements extension, thus the gain valuation formula is defined as:

$$G(s) = Q_s^l \cdot (\mathrm{T}_R - \mathrm{T}_L)\,. \tag{8}$$

The resemblance to the decision rule (4) is evident, only without the comparison part against a cache congestion price as in the original model.

2) *Efficient Mobility and Popularity-based Caching with Replacements (EMPC-R).* This model version refers to the integration of both cache replacements and requested content popularity to cache decisions, hence the gain valuation formula is defined as:

$$G(s) = (Q_s^l + w \cdot f_s^l) \cdot (\mathrm{T}_R - \mathrm{T}_L). \tag{9}$$

This formula integrates both user mobility information Q^l and popularity information f_s^l. Whereas Q^l remains as defined in (4), f_s^l is the probability that object s gets requested again from cache l in the future, weighted by a special tuning factor w. This factor is the number of requests served by l to its currently attached users during one handover interval. Hence, w adapts a dynamic value that is used to *"tune"* the balance between mobility prediction and content popularity information on gain valuation by growing the significance of popularity expressed via f_s^l with the number of requests currently served by l. We can approximate f_s^l as defined in (10), where T_{req}^l is the time duration between two consecutive requests for any object submitted to cache l and I_s^l

$$f_s^l = T_{req}^l / I_s^l\,, \tag{10}$$

is the time between two consecutive requests for s submitted to l, i.e. f_s^l reflects the contemporary popularity of s in cache l w.r.t. the most recent requests information.

3.3. Model application in Heterogeneous Wireless Networks

Heterogeneous Wireless Networks (HWNs) are characterised by the coexistence of different radio access technologies. Within the context of this paper, we assume HWNs as the one portrayed in Fig. 2, that are comprised by one macro Base Station (BS) and multiple small cell BSs. Each small BS is equipped with a cache storage and has a running instance of our model to enhance mobility support for users with *niche* requests who are primarily concerned about download charges. Such *cost-concerned* users may choose to postpone entire downloads until they connect to a small cell, or to download data concurrently from the macro and any ephemeral small cells connections. Along these lines, social networking users are a notable example: they request for content that is tailored to their personalised social network (See Section 2.2 on page 6) and they commonly choose to receive only notifications or small objects from the macro BS and to engage into data downloads of considerable size *later* via a Wi-Fi hotspot.

Fig. 2 depicts a cost-concerned user moving across the macro area and connecting to the small cells along its path. The location and the range of coverage of the small cells is up to device settings and landscape properties, while clusters of contiguous coverage such as in the figure can denote the presence of a provider-orchestrated[3] coverage. To apply our model, each running instance (see Section 5 on page 33) has to attain information about the user's (i) transition probabilities, (ii) expected handover and small-cellular "residence" session durations, (iii) content requests and (vi) the corresponding data-transfer costs. To better utilise the available backhaul and storage resources during prefetching, requests can be fine-grained to the level of the content chunks *predicted* to be consumed during the mobile's ephemeral residences.

3.4. Transfer costs integration

Our model's cache actions are designed to reduce monetary costs or download delay via trying to increase transfer cost gains $\mathrm{T}_R - \mathrm{T}_L$. *Monetary* costs may refer to user service charges for macro-cellular downloads, but also to charges and other expenses for providers, e.g. due to inter-domain data transit services used for fetching data from external sources. Delay on the other may refer to the propagation or transmission delay suffered by users when downloading their desired content, with a considerable impact on QoE. This paper adapts a user-oriented approach by addressing niche mobile requests, particularly for video, hence our following discussion is focused upon (i) user *charges* and (ii) download *delay* and its relation to QoE.

3.4.1. Monetary charges

We assume monetary charges M_{MC} and M_{SC} for macro- and small-cellular usage in HWNs respectively, thus the corresponding transfer costs adapted by our model are:

$$\mathrm{T}_L \equiv \mathrm{M}_{SC}\,, \mathrm{T}_R \equiv \mathrm{M}_{MC} \tag{11}$$

Our model can be applied only when $\mathrm{M}_{SC} < \mathrm{M}_{MC}$ such when users utilise a third-party Wi-Fi hotspot connection free of charge or in case they are offered with favourable charges[4] for connecting to their own providers' pico/femto resources as a means of motivation for increasing the amount of the offloaded traffic to small cells.

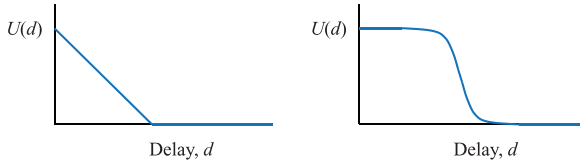
3.4.2. Delay cost

Delay implies an important cost for both users and their providers. It can cause video buffering breaks or lead to lower video qualities such as with the Dynamic Adaptive Streaming over HTTP (MPEG-DASH) protocol in order to match the offered level of QoS. Therefore, on the one hand there is delay as a metric, and on the other hand there is the users' own utility of delay which directly relates to the perceived user QoE. In the following, we provide an analytical discussion on delay w.r.t. both its measurable value and its utility. Given our focus on big objects and particularly video content, the analysis focuses on transmission delay, which is a function of the available throughput relative to the transited content volumes and the QoS requirements of the mobile application. Additionally, we present a user utility function which can represent users' valuation for delay d and via which we may capture the impact of delay on QoE:

- *Transmission delay*. Let R_W be the throughput of the wireless small cell interface and and R_{bkhl} be the corresponding throughput of the backhaul link or the path to the source of data via the backhaul. Also, assume that $R_{bkhl} < R_W$, which justifies that prefetcing and caching data can yield benefits. Without proactive caching, the delay for downloading content s with size o_s is $d \equiv d_{bkhl} = o_s/R_{bkhl}$ as the throughput to a mobile is constrained by R_{bkhl}. However, *with* proactive caching, delay is $d \equiv d_W = o_s/R_W < d_{bkhl}$. Due to the lower delay and the higher throughput levels $R_W > R_{bkhl}$ that can be utilised to transfer more data to mobiles, the users' QoE can be improved such as

[3] Philips SmartPole (with 4G LTE support from Ericsson) street lighting in Los Angeles: goo.gl/KzCsVs.

[4] www.thinksmallcell.com/Operation/billing.html.



(a) Linear utility. (b) Sigmoid utility.

Fig. 3. Example utilities as a function of delay. Users obtain *no* value above a maximum delay threshold. (a) Linear utility. Value increases as delay approaches to zero. (b) Sigmoid utility. Max value below a min delay threshold.

by transferring videos in a higher quality. If only a part x_s^l of s is prefetched to small cell l, then the remainder $o_s - x_s^l$ is compensated via the backhaul. If the mobile requesting s moves to l, then the small-cellular delay for transferring the whole of s is:

$$\mathcal{D}_{sc}(x_s^l) = \frac{x_s^l}{R_W} + \frac{o_s - x_s^l}{R_{bkhl}} = \frac{o_s}{R_{bkhl}} - \left(\frac{1}{R_{bkhl}} - \frac{1}{R_W}\right)x_s^l. \quad (12)$$

- *User utility function* $\mathcal{U}_s(d)$ Fig. 3 portrays two possible forms of $\mathcal{U}_s(d)$, namely, a *linear* utility that *decreases* with delay in Fig. 3a, and a *sigmoid* utility for which a maximum value is achieved below a minimum delay threshold in Fig. 3b. We can define utility

$$U_s^l(x_s^l) = \mathcal{U}_s\left(\mathcal{D}_s(x_s^l)/q_s^{i,l}\right) \quad (13)$$

as a function of the part x_s^l of object s that is proactively fetched in cache l. If a user has a higher probability $q_s^{i,l}$ to move to l, then it needs to proactively cache a larger part of the content to achieve the same utility. We assume that (13) is *continuous* and strictly *increasing* in $[m_s^l, M_s^l]$, where $m_s^l \geq 0$ and $m_s^l < M_s^l \leq o_s$ are minimum and maximum values of x_s^l for which $U_s^l(x_s^l) = U_s^l(m_s^l)$ for $x_s^l \leq m_s^l$, and $U_s^l(x_s^l) = U_s^l(M_s^l)$ for $x_s^l \geq M_s^l$. An example corresponding to Fig. 3a is $U_s^l(x_s^l) = \frac{\mathcal{R}}{q_s^{i,l}}(x_s^l - m_s^l)$, where $x_s^l \in [m_s^l, M_s^l]$ and $\mathcal{R} = \frac{1}{R_{bkhl}} - \frac{1}{R_W}$.

4. Performance evaluation

We present an extensive performance evaluation conducted with a custom simulator built especially for evaluating our model and model extensions in heterogeneous wireless networking application scenarios. Our performance results show the mean monetary or delay cost *gains* from small-cellular data consumption as a percentage of the cost of the corresponding pure macro-cellular communication. The presented gain percentages and their corresponding 95% confidence intervals refer to 100 simulation repeats. Apart from EMC, EMC-R and EMPC-R, we include the performances of three benchmark models, namely, Naïve , Max Popularity (MaxPop) and NoCache (Table 1).

4.1. Geospatial and wireless properties

We begin our evaluation with studying the impact of geospatial and wireless properties on performance. These properties refer to (i) the range of coverage, (ii) the mobile handover policies and (iii) the overall cache-storage *supply over demand* ratio. We adapt the following simulation setup, which is convenient at this stage for focusing our analysis exclusively on the geospatial and wireless properties of our model. Unless stated otherwise, all results presented throughout Section 4.1 refer to: (i) *topology:* stochastically distributed small cells across a 700 × 700 macro cell area; (ii) *mobility model:* "Uniform" user mobility pattern unless stated otherwise (see Table 2 for details); (iii) *cache capacity supply:* cache supply of 100 content objects per small cell; (iv) *gains upon hits:* fixed, to 90% gains upon cache hits; (v) *user demand:* 2800 mobile users initialised at random points of the area, each with a single, unique request. Since unique, requests are equally-popular and the EMC model has to consider only one mobile transition probability per object request. Thus, we use only the basic model here, leaving the popularity and legacy caching extensions for Section 4.2.

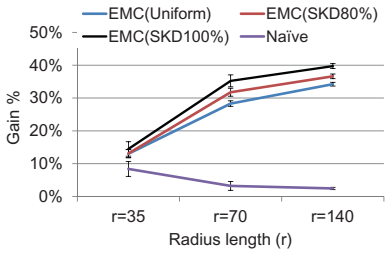


Fig. 4. Impact of user mobility on performance using 35 small cells with a short, medium or long radius (r).

4.1.1. User mobility

Fig. 4 shows EMC and Naïve gains for the mobility models of Table 2. The more skewed the mobility model is, the greater are the performance gains for EMC. This is expected due to the fact that mobility prediction becomes more accurate with increased mobility skewness, hence mobile decisions can better utilise the available buffer space resources. Nevertheless, even EMC (Uniform) manages to utilise its buffer space better than Naïve by ~6% due to the cache pricing scheme adapted in EMC. In another conclusion, the graph shows that *increasing* the *radius* length r of the small cells can significantly increase the performance of EMC. The former causes demand to increase, which leads to a better cache-storage utilisation in the case of EMC due to its congestion pricing mechanism. On the very contrary, exposing Naïve to a higher demand degrades its performance. Its gains difference from EMC range from ~6% for $r = 35$, to ~25–32% for $r = 70$ and even to ~31.8–37.3% for $r = 140$, as Naïve (i) lacks the ability to handle higher demand and (ii) neglects mobility and cache space congestion.

4.1.2. Handover policies

Fig. 5 shows EMC gains as a function of cache supply over demand for two handover policies, namely, *Cached Content (CC)* that allows mobiles to attach to cells that have prefetched their requested content, and *Closest Range (CR)*, according to which mobiles attach to the closest cell within range. The graph show that

Table 1
Cache performance benchmarks. MaxPop and NoCache used only in Section 4.2.

Notation	Description
Naïve:	Caches content to all neighbours with available cache space. Such "blind" proactive caching lacks intelligence and helps to point out the benefits of adapting user mobility or content popularity, as well as cache congestion pricing in cache decisions.
MaxPop:	Prefetches the *topmost* popular videos at small cell based on their long-term requests frequency. MaxPop lacks locality knowledge, hence it helps to highlight the advantages of adapting up-to-date user mobility or local popularity to cache actions.
NoCache:	*No* proactive caching used. Mobiles consume data *only* from the backhaul.

Table 2
Mobility models, i.e. patterns of probabilistic user mobility in space. All models integrate a ± 2.5% probability jitter. Note that each model creates the necessary conditions to approximate, yet *not* to impose, corresponding handover probabilities between cells, as this is also w.r.t. to the location and range of the cells.

Notation	Description
Uniform:	The number of mobiles moving along each direction is uniformly equal.
SKD80%:	Skewed mobility model with 80% of mobiles moving towards the same direction.
SKD100%:	Skewed mobility model with 100% of mobiles moving towards the same direction.

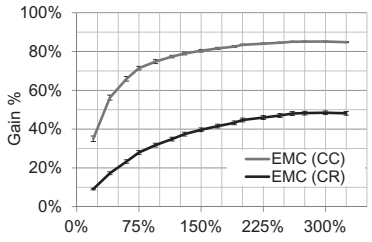


Fig. 5. Impact of handoff policies on performance. The X-axis shows the supply of storage-cache from 21 small cells as a percentage of total demand from all mobiles.

both performances continuously increase, yet with a decelerating growth rate. Performance is higher for CC by ∼37.5–42% as users maximize their gain by attaching to the cells which offer their requested objects directly from a local cache. Note though that the presented gain measurements say nothing about actual data-rates, hence *hybrid* CC/CR policies may combine cache hits and corresponding low access latencies with higher wireless rates.

4.1.3. Transient performance

The graphs of Fig. 6 show the transient performance of EMC compare to Naïve for different levels of storage supply over a fixed level of demand. Both graphs show that gains *converge* to a *steady-state*. For EMC, this happens *shortly* after the beginning of the simulation, i.e. from a initial state of *no* knowledge on information about user mobility and requests. This strongly indicates that EMC *can adapt quickly to changes* in demand and user mobility. Furthermore, graph Fig. 6a shows that doubling supply S over demand D from 25% to 50% nearly doubles average gains from ∼36% to ∼65.5%, whereas further doubling S/D to 100% or 200% corresponds to a ∼13.5% and ∼5% of extra cost gains on average. Evidently, cache expansion can yield significant performance benefits in the case of EMC, particularly, when increasing small capacities. On the very contrary, cache expansion offers minor extra gains to Naïve. As portrayed in graph Fig. 6b, transient performances for Naïve overlap for $S/D \geq 50\%$, being broadly lower than EMC. Recall that Naïve's performance in Fig. 4 shows its inability to handle higher demand due to neglecting mobility and cache space congestion. Likewise, here we observe that it can not exploit higher cache buffer supplies for the same reason.

4.2. Mobile video gains

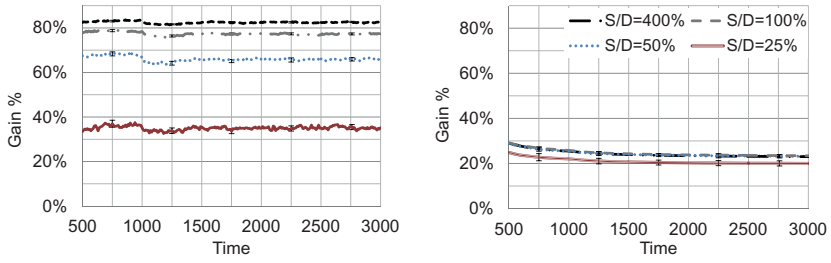
In this second part of the evaluation we provide a meticulous performance study w.r.t. user charges and delay gains in scenarios that involve mobile video demand. Unless stated otherwise, the presented results correspond to the following default setup:

1) *Wireless throughput:* Mobiles have a constant 2 Mbps access to the macro-cell and an extra 4 Mbps of wireless throughput when linked to a small cell. Users can also leverage up to 2 Mbps from their hosting small cell's backhaul capacity, i.e. mobiles can jointly download data from the local cache and the backhaul up to 4 Mbps during their ephemeral small-cell sessions. We also study performance w.r.t. different macro-cellular throughput values in Section 4.2.1.
2) *Content requests:* We use a set of five different synthetic traces produced with the GlobeTraff [25] workload generator, which comply with the latest literature modelson requests popularity and temporal locality. Each trace refers to a different Video Catalogue (VC) of 100K videos with average file size ∼100 MB, split into ∼2.5 MB chunks. We adapt s=0.75 after literature observations [10] as the default value of the Zipf distribution exponent parameter used for video popularity. For completeness, we also study performance w.r.t. (i) a series of different s values in Section 4.2.2, and (ii) real web traffic[5] after filtering-out non-video requests in Section 4.2.3.
3) *Topology & user mobility:* We use 22 small cells stochastically distributed over a 1000 × 1000 area and a real mobility trace [26] with GPS data of 536 taxi cabs over a period of 30 days in the San Francisco Bay Area, USA. The chosen small cell density corresponds to data about the area extracted from the publicly available WiGLE database[6] and after grouping Wi-Fi hotspots within less than a 100 m distance from each other. Besides the default setup, we adapt different mobility traces in Section 4.2.4 and study the impact of alternative densities in Section 4.2.7.
4) *Mobile & Stationary demand:* Mobiles have 1 active request at a time which is *jointly* served via the macro and the small BSs. During a mobile's *"residence"* in a small cell, it consumes data from the local cache in parallel to consuming the non-cached video chunks from the small cell's backhaul and the macro link. Apart from the mobiles, small cells also host "stationary" devices with long-lasting connections, e.g., laptops. We assume an average number of 20 active stationary requests in each target small cell, which last for a time that is equal to the average mobile *handover time* to the small cell. Also, we study the impact of the number of stationary requests in Section 4.2.6.
5) *Cache capacity:* We assume 4 GB cache-storages for simulating a low-cost, highly-distributed HWNs along the lines of Fig. 2. Also, we compare the performance of our model against different cache storage capacities in Section 4.2.8.
6) *Cost gains:* Macro-cellular (resp., small-cellular) transfers cost 10 (resp., 0) monetary units per downloaded data unit. Note though that the results presented in Section 4.2.10 adapts a series of cost combinations which refer to transfer delay.

4.2.1. Performance against benchmarks and macro-cellular throughput

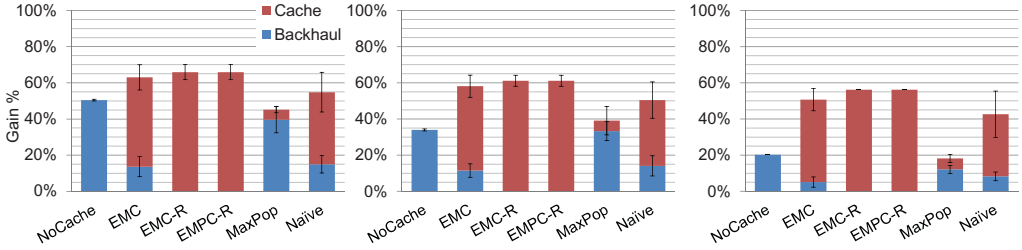
Fig. 7 shows performance for 3 different wireless macro-cellular throughput values MC_t: (i) low throughput (Graph Fig. 7a); (ii) average throughput (Graph Fig. 7b); and (iii) high throughput (Graph Fig. 7c). As a general comment, we observe that all performances drop with MC_t because the mobiles consume an increased part of their requests from the macro cell during both their han-

[5] http://ita.ee.lbl.gov/html/traces.html.
[6] Wireless Geographic Logging Engine: https://wigle.net/.



(a) EMC (b) Naïve

Fig. 6. Transient performance for different supply over demand ratios (S/D).



(a) $MC_t = 1\ Mbps$ (b) $MC_t = 2\ Mbps$ (c) $MC_t = 8\ Mbps$

Fig. 7. Performance with different macro-cellular wireless throughput levels. Graphs portray a performance breakdown between download gains from the cache (in red) and from the backhaul link (in blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

dover and residence periods. The extend of the drop is better perceived via observing NoCache: its performance falls from 50% to 34% and from there to 20% in correspondence to a 2× and an 4× MC_t increase respectively. The impact on cache gains is less for EMC (∼−1.5%) with each MC_t increase compared to the extended model variations, whose gains fall from 66% to 61% and 56% respectively. Last, cache gains for MaxPop are robust but low (6%), and Naïve cache gains drop by 4% and 2% for each corresponding MC_t increase. Next, we discuss the performance against the benchmarks. Without loss of generality, we focus on the average-throughput scenario of Graph Fig. 7b, which is the default scenario (see Section 4.2); yet the same conclusions apply to the rest scenarios as well:

1) *Maximum gains from cache with replacements:* EMC-R and EMPC-R appear[7] to exploit small-cellular wireless throughput exclusively with data straight from the local cache, due to the added efficiency of cache replacements that yields ∼14% more cache gains relative to EMC.
2) *EMC yields a robust and overall good performance:* The difference between EMC and EMC-R/EMPC-R is *only* 3% when including backhaul gains. This outcome is *highly important* as cache replacements imply a considerable computational overhead relative to the basic model (see Section 5.4 on page 14).
3) *EMC and EMC-R adapt well to temporal locality:* Even though only EMPC-R directly addresses content popularity, EMC and EMC-R decisions appear to adapt well to contemporary popularity conditions via the aggregated transition probabilities Q_s^l of the mobiles with an active request for content s. Such *short timescale*[8] and dynamic *mobility information* help to adapt quicker and better to temporal locality conditions than long-term popularity information. A comparison against the cache gains of MaxPop (6%), verifies that EMC corresponds better to temporal locality than the long-term popularity used by MaxPop, thus exceeding the latter's performance by 41% (19% including the backhaul). Also, EMC-R and EMPC-R appear to have the same performance. This does *not* imply that popularity information in (9) is needless, as it allows to decide upon legacy-cached objects with the cache replacements extension.
4) *Intelligent Vs. naïve caching:* Naïve has 11% less cache gains (8% including the backhaul) relative to EMC. This fair performance difference is due to (i) the total user *demand* of the taxi trace, which is generally *not* high. Many out of the simulated taxi cabs can have long and varying handover periods that make them consume most (or even the whole) of their requested content before entering a small cell. Moreover, (ii) Naïve decisions are not totally "blind" and regard only a part of the requested contents (see Section 5.2.2 on page 36), while the gains of intelligent caching can be much higher against a "pure" Naïve as we show in Sections 4.1.1 and in 4.1.3. In addition, (iii) Naïve's gains come with larger confidence intervals, which implies that EMC and especially EMC-R/EMPC-R are less susceptible to demand fluctuations.
5) *Compared to NoCache:* Gains from mobility-based proactive caching are significant compared to using only the backhaul. EMC (resp., EMC-R/EMPC-R) outperform NoCache by ∼24% (resp., 28%), while MaxPop by only 6% and Naïve by 16%.

4.2.2. Content popularity skewness

Fig. 8 shows the gains of our model as a function of the Zipfian video popularity distribution exponent parameter (s). We use a fixed 100 MB size for all videos so as to focus exclusively on the impact of content popularity *skewness* as it increases with s. The results show a robust performance for the model extensions and an increasing trend for EMC. EMC better utilises the cache with more skewed content popularities (greater s), which is expectable as cache decisions regard mostly specific content. Also, its performance approaches to that of the model extensions, being only ∼7.6% lower for $s \geq 0.9$. Note, however, that our model targets niche requests; highly popular content can be anyway addressed by CDNs (see Section 4.2.11).

[7] Given this setup. See also Section 4.2.9. for a more meticulous analysis w.r.t. different system parameters.

[8] An outcome which largely coincides with [4], which concludes that "We need to take into consideration the latest mobility information from nearby devices to make accurate predictions" w.r.t. urban environments.

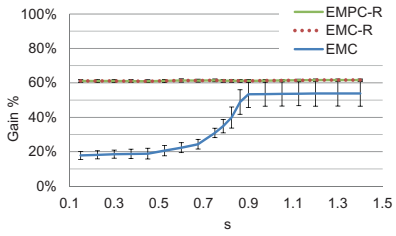


Fig. 8. Performance against an increasing Zipf exponent parameter *s*. Content popularity skewness increases with *s*.

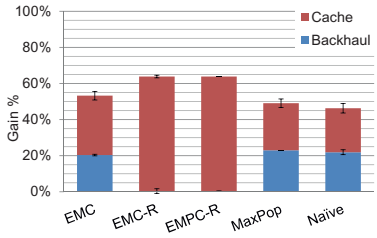


Fig. 9. Performance using real video requests. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

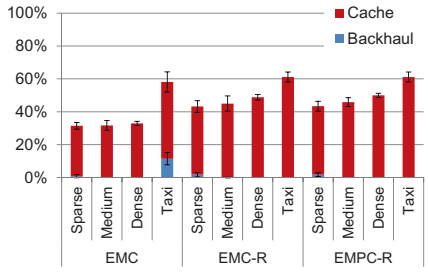


Fig. 10. EMC and model extensions gains for the taxi cab [26] and KTH/Walkers [27] mobility traces. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4.2.3. Real trace requests

Fig. 9 presents gain performance w.r.t. a real trace of web requests[5]. A comparison between Graphs Fig. 7b and 9 reveals that EMC gains are lower than with GlobeTraff: Cache gains are less by 14% (53% including the backhaul). Opposite to EMC, the gains of the extended models are increased by 2%. Gains for Naïve also drop, remaining lower by 9% relative to EMC. Interestingly, MaxPop's cache gains are 20% higher (10% including the backhaul) compared to Graph Fig. 7b, along with smaller confidence intervals. Moreover, they exceed Naïve's cache gains by 2% despite being 30% lower than Naïve's in Graph Fig. 7b. This dramatic increase can be due to an increased popularity skewness and/or due to the size of the pool of video files (see Section 4.2.9 for more).

4.2.4. Walkers mobility traces

We use the "KTH/Walkers" dataset [27] as a benchmark to the results with the taxi cab mobility trace and present the results in Fig. 10. We adapt the same small cell density as with the taxi trace, only for a 400 × 400 area to which the KTH/Walkers traces refer to. The walkers trace comes in three versions w.r.t. the density of users in space: (i) "sparse", (ii) "medium" and (i) "dense". User density with the taxi trace is stable and closer to "medium" relative to the rest. However, density can vary with time with KTH/Walkers. In addition, walking speeds are less variable than the taxi driving speeds which can be anywhere between very slow (e.g., due to traffic jams or traffic lights) and very high.

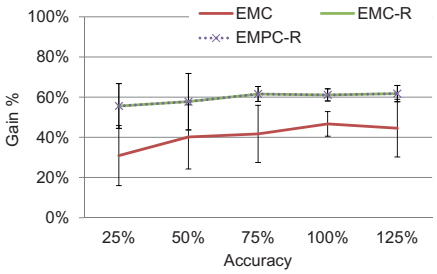


Fig. 11. Performance against different mobile prediction accuracy levels. Assume handover time *h*. Accuracy *x* on the x-axis denotes that mobile prediction considers time $x \times h$, i.e. the performance for $x = 100\%$ corresponds to the most accurate prediction. The same applies w.r.t. the predicted residence time.

These differences lead to a smaller level of cache demand with KTH/Walkers, either because the active users are at times less, or because the handover or residence periods last less, which implies that less data can get prefetched to the caches or consumed while being connected respectively. As a result:

1) *EMC cache gains are lower:* They drop by ∼15% and there are no backhaul gains as the residence in the small cells lasts too little to utilise the backhaul link.
2) *User density appears to have small impact:* There is no impact on EMC and only a small one on EMC-R/EMPC-R whose gains increase from 42% to 45% and 47% for "sparse", "medium" and "dense" respectively. Interestingly, there are some backhaul gains (0.5–1.7%) for "sparse". The corresponding cache gain differences from the taxi trace are: 19% (∼17% including the backhaul), 16% and 12%.

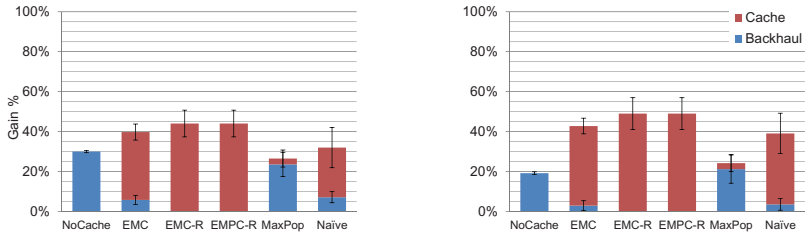
4.2.5. Mobility prediction accuracy

As discussed in Section 5.2.2 on page 36, the prediction of the *consumable* part of a mobile's content request is an integral part of the cache allocation process that is based on the mobile's predicted handover and residence duration times. To clarify the impact of these mobility predictions, we present the graph of Fig. 11 which shows the performance against different accuracy levels. As explained in the legend, prediction is most accurate for $x = 100\%$, while it can lead to buffer underutilisation (resp., waste) for $x < 100\%$ (resp., $x > 100\%$). Indeed, the graphs shows that gains increase with accuracy and that the best average performance along with the smallest confidence intervals corresponds to $x = 100\%$. There is though a difference between the basic and the extended models. The latter converge earlier and preserve their performance for $75\% \leq x \leq 125\%$ due to cache replacements, while gains for EMC are lower and the corresponding confidence intervals are more than twice larger for $x \neq 100\%$.

4.2.6. Stationary user requests

Fig. 12 shows the aggregate performance for all users' requests. The total number of connected mobiles per small cell converges to ∼7 with time for both graphs; yet the average number of stationary requests is set to 21 (resp. 7) for Graph 12 a (resp. Graph 12 b). Our conclusions are summarised as:

1) *Less backhaul gains for less stationary requests:* Stationaries have less hits as cache actions are based on mobiles' requests. Thus, they consume more data from the backhaul than the mobiles, which explains the reduced backhaul consumption in Graph 12 b by 10.8%, 2.9% and 3.6% for NoCache, EMC and Naïve respectively. Note MaxPop's backhaul decrease by 3% and the higher impact on NoCache because of the proportionally more mobiles in Graph 12 b that consume data also from the macro cell.



(a) #MU25% (b) #MU50%

Fig. 12. Overall performance w.r.t. requests by both mobile and stationary users. Graph 12 a (resp., 12 b) corresponds to 21 (resp., 7) stationary user requests, i.e. 25% (resp., 50%) of the total number of all downloads accounts for mobile requests. Cache decisions are the same in both graphs due to using the same input information: requests, transition probabilities, handover and residence times, long-term popularities. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

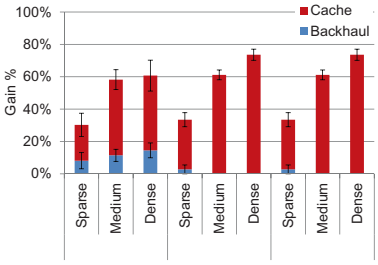


Fig. 13. Performance against different small cell densities. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

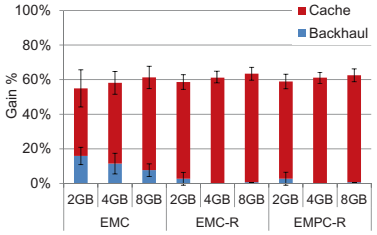


Fig. 14. Performance against different levels of cache storage supply. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2) *Cache gains increase for less stationary requests:* Cache gains are higher in Graph 12 b by 5.9%, 5.0% and 10.6% for EMC, EMC-R/EMPC-R and Naïve respectively, due to the increased proportion of mobiles. The former outweigh corresponding backhaul loses, hence the aggregated backhaul plus cache gains increase by 3%, 5% and 7.2%. Note that MaxPop cache gains remain approximately equal to 3%.
3) *EMC-R and EMPC-R have the same performance:* Caching with EMC-R takes content popularity into account via $w \cdot f_s^l$ in formula (9), where w tunes the impact of popularity based on the number of requests served to attached users during a handover interval. Thus, w adapts a different value in the two graphs, but only causes a marginal gain difference compared to EMC-R, verifying our former conclusion (Section 4.2.1) that the model can adapt well to temporal locality via *only* mobility information.

4.2.7. Small cell density

Fig. 13 shows performance gains for three different densities over the 1000 × 1000 simulation area: (i) "sparse": 11 small cells; (ii) "medium": 22 small cells; and (iii) "dense": 55 small cells. Increasing the density increases cache gains in a twofold way: First, because it increases the available cache supply and, second, because it increases (resp., decreases) the aggregate residence time in small cells (resp., handovers) during which the mobiles can download more cached or backhaul data from the small cells (resp., the mobiles can download only from the macro cell). To sum up:

1) *Cache gains increase with cell density:* Doubling the number of small cells from sparse to medium increases 2.14× and 1.96× the cache gains of EMC and EMC-R/EMPC-R respectively. Further increasing density from medium to dense by 2.5× has no impact on EMC cache gains but it adds up 12% to the cache gains of EMC-R/EMPC-R.
2) *Backhaul gains increase with density only for EMC:* Backhaul gains for EMC increase with density from 8% to 11%, and to 15%. However, the few backhaul gains (∼3%) with a sparse density for EMC-R/EMPC-R cease to exist. While a greater residence duration helps EMC to cover up for cache misses with backhaul data, it helps EMC-R/EMPC-R caching to become even more efficient and not need the backhaul.

4.2.8. Cache storage supply

Fig. 14 on page 29 portrays the performance of our model variations for three different cache storage sizes in each small cell: (i) 2 BG, (ii) 4 BG and (iii) 8 BG. The results have similarities with Fig. 13 due to increasing the available cache supply, yet with less significant added cache gains. Our main conclusions are as follows:

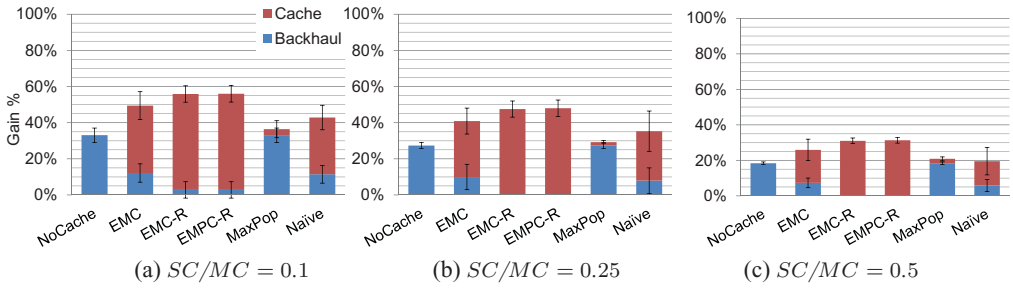
1) *Cache gains increase with storage size:* Doubling the storage buffer from to 4 GB increases EMC cache gains by 8% and EMC-R/EMPC-R by 5%. Further doubling to 8 GB adds up 7% to EMC cache gains and only 2% to EMC-R/EMPC-R.
2) *Declining backhaul gains:* Backhaul gains for EMC drop from 16% to 11%, and from 11% to 8%. Likewise, any backhaul gains (∼2.5%) that exist with the 2 GB scenario for EMC-R/EMPC-R cease to exist because increasing storage improves cache hits, thus decreasing the need for utilising the backhaul to cover up for cache misses.

4.2.9. Video catalogue size and mobile demand

Table 3 shows the average gains for two mobility-based caching demand over total cache storage ratios: low (0.6) and high[9] (6), and two Video Catalogue (VC) sizes in terms number of files: small (∼10.5K) and large (100K).

1) *Caching demand impacts performance significantly:* In the examined scenarios, the impact of mobiles' caching demand is higher than that of |VC|. Increasing |VC| affects confidence intervals, increases gains for EMC-R/EMPC-R, but decreases the other performances, particularly MaxPop's: given the fixed-sized 4GB buffers, MaxPop caches a diminishing fraction of the top-most popular videos with growing |VC|.

[9] High caching demand corresponds to x10 video file size increasebut the exact impact on the demand level depends on mobility, i.e. on the handover and residence times as discussed in Section 5.2.2 on page 36.



(a) $SC/MC = 0.1$ (b) $SC/MC = 0.25$ (c) $SC/MC = 0.5$

Fig. 15. Performance for different small-cellular over macro-cellular delay cost ratios (SC/MC). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Cache gains w.r.t. a low and a high video caching demand over total cache storage ratios (Dmd), combined to a low or large Video Catalogue (VC). Parentheses contain confidence interval values. Line 2 marked with (∗) corresponds to the default setup explained on page 22. All results refer to the same local cache buffer size (4 GB) and the same total cache storage supply (22 small cells × 4 GB).

Dmd	|VC|	EMC	EMC-R	EMPC-R	MaxPop	Naïve
Low	Small	47.7% (5.8%)	60.4% (2.2%)	60.4% (2.1%)	11.7% (12.1%)	37% (8.3%)
*Low	Large	46.7% (6.1%)	61.1% (3%)	61.1% (2.8%)	5.8% (7.9%)	36.4% (10.1%)
High	Small	10% (4.4%)	26% (5.1%)	26.1% (4.6%)	1.5% (1.7%)	5% (3.4%)
High	Large	10.4% (8%)	38.9% (8.7%)	38.9% (8.5%)	0.5% (0.3%)	4.9% (5.2%)

2) *Maximum gains from cache replacements:* Given a low caching demand, cache replacements add up ∼12.5% to performance relative to the basic EMC model for either a small or a large VC. The impact is even more significant w.r.t. a high caching demand, yielding 2.6× (resp., ∼3.7×) the gains of EMC with a small (resp., large) VC.
3) *Good gains from mobility prediction:* Assuming low caching demand scenarios, gains are high from mobility prediction based on small timescale information, especially for a larger |VC|: EMC yields 4.1× (resp., 8.1×) the gains of MaxPop for a small (resp., large) |VC|. Likewise in high demand scenarios, EMC yields 6.7× (resp., 20.8×) the gains of MaxPop and 2× (resp., 2.1×) that of Naïve for a small (resp., large) |VC|.
4) *Easier to benefit from requests' popularity with*low*caching demand:* All gains, *particularly MaxPop's*, are high relative to the corresponding gains for a high caching demand. Since all setup combinations in the table refer to the same mobility model, the former indicates that we can better exploit requests' popularity with a smaller demand.
5) *Low gains, particularly from popularity, with high caching demand:* Gains are generally low with high cache demand, especially for Naïve and MaxPop. Corresponding gains for EMC are ∼4.8× (resp., ∼4.5×) lower relative to its own performance in scenarios that combine low demand with a small (resp., large) |VC|.

4.2.10. Delay gains

We complete our evaluation by studying our model's performance from a download *delay* perspective. We adapt the same setup that is explained on page 22, with the difference of assuming transfer costs which reflect users' cost valuation of download delay. Moreover, we try to approach the impact of CDN presence. Despite its growing scale, the Internet works thanks to CDNs. However, the current state of the art does not consider the possible implications caused by CDN content replication on local caching performance. Opposite to that, we try to study the impact of a CDN on local caching in support of our discussion in Section 2.3 on page 8, where we argue for complementing CDNs by targeting niche requests locally.

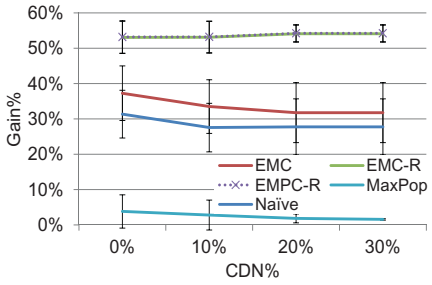


Fig. 16. Cache gains against the percentage (CDN%) of the topmost popular videos cached by CDNs.

Fig. 15 shows gains for 3 different small-cellular over macro-cellular delay cost ratios *SC*/*MC*, each of which can reflect how users assess, e.g. based on QoE measurements, the benefit of using a small cell relative to pure macro-cellular usage: (i) low ratio, high benefit (Graph 15 a); (ii) average ratio and benefit (Graph 15 b); and (iii) hight ratio, small benefit (Graph 15 c). As a general comment, all performances, either w.r.t. cache hits or the backhaul, drop with increasing *SC*/*MC*. Also, the gain differences between the different cache models also tend to decrease. This is normal to expect as the corresponding gains from caching also drop from 90% to 75% and 50% for the presented cost ratios in the figure respectively. As with other graphs, we observe that EMC-R and EMPC-R share the same performances, which are higher than EMC and any other benchmark. EMC in particular, has a robust gain performance relative to the extended model versions. Its corresponding cache gain difference from the extended models grows smaller from ∼16% to 12% with increasing cost ratios. Moreover, this performance difference is significantly lower when considering also the gains from the backhaul: ∼5–6.5%. As we also highlight in Section 4.2.1 on page 23, this is important due to the computational overhead of the cache replacements extension. In addition, we observe that the cache gains from MaxPop are robust but low, and that all the caching models have benefits over NoCaching. For instance, cache gains from EMC are higher than from NoCache; if backhaul is further included, then the added gains from EMC are 14%, 8% and 7% for the portrayed cost ratios respectively. Last, the further gains from EMC relative to naïve are fair and robust: ∼5–7% (resp., ∼5.5–7.5% with backhaul).

4.2.11. The impact of CDNs delay gains

Next, the results of Fig. 16 correspond to a scenario of *SC*/*MC* = 0.1, where CDN presence reduces the transfer delay costs of the top-most popular videos (CDN%) that it serves by 75%. Notice that

Table 4
Notation.

Notation	Descripion
l:	A handover destination (target) small cell.
i:	Origins (source of handover) small cell.
CDM_x:	The CDM running at small cell x.
s or s_c:	A requested content or the c-th chunk of the content.
M_i:	A set of mobiles attached to some origins cell i.
N_i:	i's neighbourhood, i.e. set of next hop destinations from i.
I_l:	The set of the different origins cells from where mobiles can move to l.
$\hat{M}_l$:	The set of all mobiles that can move to l.
$\bar{r}$:	The average number of content requests per mobile.
$\bar{r}_c$:	The average number of requested chunks per mobile request r.

cache gains for EMC-R/EMPC-R are *robust*. This is due to the cache replacements extension which identifies what content is mostly beneficial to preserve in the cache. Contrarily, *popularity*-based caching appears to have a significant *loss*. Gains in general decrease with CDN% due to the reduced transfer costs offered by the CDN level. The performance of MaxPop in particular implies that with merely 20% of the top-most popular videos being addressed by the CDN level, there is very little to gain from popularity-only caching. Likewise[10], EMC gains converge for $CDN\% > 20\%$. Only unlike MaxPop, EMC continues to yield cache gains due to capturing content for the niche requests missed by the CDN level.

5. Implementation

Our model employs a distributed design which can be incarnated via special Cache Decision Modules (CDMs) running in small cell BSs of HWNs such as portrayed in Fig. 2 on page 16. CDMs are responsible for orchestrating cache actions after receiving and processing information about mobiles' requests, and by maintaining a distributed state via exchanging control messages within their neighbourhood. In what follows, we present 4 basic CDM functions, namely, (i) user requests, (ii) mobility prediction, (iii) content popularity adaptation and (iv) cache replacements, from an implementation perspective along with a corresponding analysis of the implied computational, memory and intercommunication complexities costs. The latter two functions are applied only when adapting the corresponding model extensions, hence they imply a further implementation and running complexity compared to the basic model. Apart from the analytical approach which follows next, note that in practice the implied costs depend on the level of integration in the network "stack". CDMs can *best integrate* with the network layer of Information Centric Networks (ICNs) to directly exploit named requests and other network primitives [19], particularly multicast communication from/to publisher/subscribed mobiles and CDMs; albeit the solution can be also implemented as an application over standard IP. Next, we discuss the implementation details and the corresponding complexity analysis using the notation of Table 4.

5.1. User requests

Users and content must be identified by an ICN name, URL or IP address, used for proactively fetching the objects via the backhaul network to the caches, as well as indices for maintaining a required state (see Section 5.2–5.4) for cache requests. While IP addresses have a fixed size, URLs or ICN names can be arbitrarily long, unless using a corresponding hash value such as a 20 byte-long (160 bits) SHA-1 cryptographic hash. The messaging cost for submitting such data to each neighbour of source i via unicast messages is $O(|M_i| \times |N_i|)$. However, it can be significantly reduced to $O(|M_i|)$ via multicasting each mobile's requests to all member of N_i. In any case, the corresponding memory cost in a target cell l depends on $|\hat{M}_l|$ and $\bar{r}$, i.e. it is $O(|\hat{M}_l| \times \bar{r})$.

5.2. Mobility prediction

CDMs must have good and timely information about the mobiles' transition *probabilities*, handover and small cell residence *duration times* to take accurate cache decisions. Within this content, we identify the following 3 alternative mechanisms:

1) *Centrally coordinated prediction*. A neighbourhood of small cells includes all the possible next-hop transition APs that mobiles can attach after leaving their origins small cell. Neighbourhoods adapt a static configuration and cooperate with the a macro cell which can track user mobility within its coverage.
2) *Decentralised prediction*. A fully distributed and decentralised approach involves mobiles notifying their possible destinations about their origins, allowing for a lightweight distributed neighborhood discovery and probability estimation.
3) *External mechanisms*. Transition probabilities can be attained through prediction [28] and tracking [29] mechanisms. Extracting user-specific mobility patterns is feasible with the cooperation of mobile providers or by using common navigation services such as Google Maps, to which users can explicitly declare their route and destination.

5.2.1. Memory & messaging

Complexity costs are low when using navigation services or with centrally coordinated prediction (e.g., due to the wireless MAC layer multicast ability of the macro BS). Decentralised prediction on the other requires some communication, computations and state maintenance in neighbouring CDMs: Assume a target l for mobiles requesting s, each of which corresponds to its own origins i. CDM_l maintains a small state per each i for tracking the number of incoming handovers $h_{i,\,l}$ from i and the total number of outgoing handovers H_i from i. The former are used to predict the *probability* of future mobile transitions from each i to l as defined in (14). Likewise, a history of handover and residence *duration* times from past handovers per each i can be kept. We refer collectively to all the former as "*mobility prediction information*".

$$q^{i,l} = h_{i,l}/H_i. \tag{14}$$

Noteworthy, all mobility prediction information imply *lightweight* messaging and memory costs. $h_{i,\,l}$ and H_i can be retrieved either from incoming mobiles from i or from the mobiles' requests submitted to l. Either way, there is no extra messaging implied as the needed information can be retrieved by l via piggy-backing 2 bytes for $h_{i,\,l}$ and another 2 bytes for H_i in existing communication for either the mobiles' attachment to l or for the requests submission to l (see Section 5.1) respectively. Though small, this amount of control data is reasonably big enough to

[10] Recall that EMC can adapt well to temporal locality via Q_s^l.

correspond to the locally maintained counters in memory which do not have to be bigger than 2 bytes as they must be periodically reset or readapted (e.g. with exponential smoothing) to reflect the currentness of the predicted transition probabilities. Given that $|I_l|$ counters are needed to maintain $h_{i,\,l}$ and another $|I_l|$ to maintain H_i for each i, the memory cost for transition probabilities in each CDM_l is $O(|I_l|)$. The same messaging and memory complexities, only w.r.t. 4 byte[11] long counters, apply to handover and residence duration times information, which can be maintained after past mobile transitions and residence sessions. Mobility prediction can be more *accurate* if the mobiles update l about their status (e.g. GPS coordinates) while in transit to l. This comes though with a tradeoff in terms of messaging, $O(U_f \times |\hat{M}_l|)$, where U_f is the average frequency of updates per mobile sent to l.

5.2.2. Cache decisions

CDM_l can take cache actions with each incoming request for some content s by integrating the predicted $q_s^{i,l} \equiv q^{i,l}$ from all the mobiles with an active request for s, along the lines of formula (5) as $Q_s^l = \sum_{\forall i} h_{i,l}/H_i$. By further leveraging the knowledge of macro/small-cellular wireless and backhaul throughputs, as well as the predictions about the handover and residence times for mobiles, CDM_l can take more *accurate* and cache-*efficient* actions at the level of separate chunks s_c. The reason lies in the fact that different mobiles can to consume different chunks of the same s from l, due to the different number of: (i) chunks κ that each individual mobile will already have consumed from the macro link before connecting to l; (ii) chunks λ that CDM_l will to be able to prefetch and cache via its backhaul while the mobile is in transit; and (iii) cached chunks ν that the mobile will be able to consume during its residence in l via its wireless interface. The earlier two are based on information about the mobile's predicted handover duration and the latter based on information about the mobile's residence duration. Hence, the chunks predicted as *consumable* from l and for which CDM_l must decide, are those en $[s_{\kappa+1}, s_{\kappa+\rho}]$, where $\rho = \min(\lambda, \nu)$. Accordingly, we adapt the predicted mobile transition information, this time per chunk, as $q_{s_c}^{i,l} \equiv q^{i,l}$ in formula (5), i.e. $Q_{s_c}^l = \sum_{\forall i} q_{s_c}^{i,l} = \sum_{\forall i} h_{i,l}/H_i$.

Based on all above, the practical cost of cache actions at l is subject to the granularity of requests. The computational complexity of each cache decision is $O(1)$ according to rule (4) and there are $\bar{r} \times \hat{M}_l$ decisions to make by CDM_l. However, if decisions consider chunks rather than whole objects, this rises to $\bar{r} \times \bar{r}_c \times \hat{M}_l$. Note that the former hold for final and irrevocable cache decisions taken upon receiving a mobile's request(s). We discuss the computational complexity cost of cache decisions again when analysing the impact of model extensions.

5.3. Content popularity adaptation

The popularity extension (see notation and definitions on page 15) implies no extra messaging costs, again $O(1)$ computations per request with an added small burden for computing $f_s^l = T_{req}^l / I_s^l$ on the fly, and a low memory cost which we analyse next: The memory cost of T_{req}^l is constrained to only a 4 byte long counter for keeping the time difference between the latest two consecutive requests. For I_s^l, we need to have $O(|S|)$ of such 4 byte long counters, where $S = \cup\{s\}$ is the set of all the different objects requested to be cached by l. Each of these counters has to be mapped to the corresponding content or chunk names which –as mentioned above– can be 20 byte long hash values. Assuming an example of 100 requests for different contents, the total memory requirement is merely $(20 + 4) \times 100 = 2400$ bytes, i.e. less than 2.5 KB. If these contents correspond to big objects, e.g. videos, which are split to ξ chunks, then the former requirement raises by a factor of ξ, e.g. less than a quarter of a megabyte for $\xi = 100$. Finally, maintaining w requires only 4 bytes for keeping the average handover duration from any source cell to l, and another 4 bytes, at most, for the number of requests currently served with data via l's cache or backhaul.

5.4. Cache replacements

Extending our model with cache replacements has no impact on messaging, yet it does have a *significant* impact on the cost of cache decisions. The cost of cache decisions is dominated by the cost of maintaining a gain-based ordering of the cached objects. This can be done with a *heap* in memory, which implies an $O(1)$ cost in terms of space and $O(n \times \log(n))$ w.r.t. time complexity, where n is the number of objects in the cache. Nonetheless, mobility prediction and content popularity information are dynamic, which implies that the gain-based ordering of all objects in the cache, as well as the gains of the pending requests, must be *periodically updated* to reflect changes w.r.t. temporal locality. This implies that the gains for all n objects must be re-evaluated and re-inserted to the heap, thus raising time complexity to $O(n^2 \times \log(n))$.

6. State of the art

The work presented in paper falls within proactive approaches for HWN environments, and approaches which use cache replacement techniques in order to approach an optimal allocation of the available cache resources. Next, we discuss the state of the art in proactive caching and cache replacement solutions.

6.1. Proactive caching solutions

Recent research [4–8] and industry[1] developments adapt proactive caching of *popular* content in small cells, as a remedy for backhaul bottlenecks. Unlike "traditional" reactive caching after users' recent requests, the idea tracks its roots to the 1990's *pre-fetching* approaches explored as delay performance enhancements for file-systems and Web access, and more recently it has regained attention for vehicular Wi-Fi access [30] and for enhancing seamless mobility support for delay-sensitive applications in publish-subscribe networks [31,32].

6.1.1. Pushing content to mobiles

Malandrino et al. [33] propose proactive seeding¥ (pushing) of content to mobiles in order to minimize the peak load in cellular networks based on a content-spreading prediction over SNSs approach. Bastug et al. [6] exploit context-awareness and SNSs to predict the set of influential users to which they proactively cache strategic contents in order to be further disseminated to their contacts via Device-to-Device (D2D) communication. In the same work, the authors also explore proactive caching to small cells during off-peak hours based on popularity, correlations among users, and files patterns. Likewise, the work in [34] by Gonçalves et al. uses D2D communication to exploit predictable demand with proactive data caching and in order to minimize user payments by trading proactive downloads. The solution yields mutual benefit for carriers via dynamic pricing for differentiating between off-peak and peak time prices. Last, *centrality* measures for content placement is used in [35], a game theoretical formulation of the data placement (caching) problem as a many-to-many matching game is given in [36], and proactive caching with perfect knowledge of content popularity in [5].

[11] Assuming Unix time for simplicity. It can be further reduced as it is used for time differences.

6.1.2. Caching in small cells

Golrezaei et al. [8] suggest proactive caching as a solution for offloading wireless traffic from a macro BS to special, femto-like "helpers", each equipped with a large cache-storage and high wireless Wi-Fi capacities. Cheung et al. [7] focus also on Wi-Fi emphasising on *delay-tolerant* applications, and propose a dynamic programming optimal delayed offloading algorithm, with the objective of *minimizing* the total cellular usage and penalizing deadline violation. A more recent work presented in [37], the authors focus on storage-bandwidth tradeoffs using the probability of not satisfying requests over a given coverage area as a function of signal-to-interference ratio, cache capacity, small cell density and content popularity.

We investigate procedures that exploit mobility prediction and proactive caching to Wi-Fi hotspots to enhance data offloading for delay tolerant and delay sensitive traffic [38,39], as well as for video streaming [40] assuming that mobiles' routes are known. In addition, our work in [41] evaluates throughput prediction to prefetch video data in integrated mobile and Wi-Fi networks in order to improve mobile streaming. Our past work on Efficient Proactive Caching (EPC) [42,43] introduces a fully distributed cache model for reducing data transfer *delay* in Publish/Subscribe (Pub/Sub) mobile network environments with *no* fallback connectivity such as a macro cell, where mobiles experience (possibly long) disconnections periods during handovers.

Our current work extends the EPC decision model and expands its application to HWNs (see Section 3.3 on page 16). From a technical point, EPC exploits *individual* user requests and individual mobility information for cache actions, whereas here we use aggregated mobility and content popularity information to further capture the dynamics of local content popularity conditions. The resulted model is tailored for addressing niche mobile demand whereas EPC was design for reducing delay and especially propagation delay of small pieces of content. Opposite to that, the current work is best suited for reducing mobile charges or delay for bigger objects such as videos, possibly under the joint use of a macrocellular download connection.

Compared to the rest of the state of the art in proactive approaches, this paper introduces a distributed model for cache decisions that can adapt quickly to changes in the mobility or the demand model (e.g., flash crowds) based on *dynamically tuned* content popularity and mobility prediction information. Furthermore, our model does not simply aim on popular content as the rest of the approaches do, but exploits mobility and individual requests information to capture *niche* demand. Even though niche requests for large objects such as videos imply a higher cache space requirement compared to caching only popular content, still our approach utilises cache-storage efficiently with its dynamic congestion pricing scheme. Last, the most striking difference of our model regards the *problem formulation*. The rest of proactive solutions use mathematical optimisation approximation for the non-tractable distributed cache problem of "content placement" [44], which is also recognised as NP-hard in [8]. These solutions are centralised, neglect mobility or employ a static adaptation of popularity that is *not* conducive to capturing the dynamics of temporal locality. Even the work in [4] that does consider mobility, *statically* splits storage for popularity- and mobility-based caching.

6.2. Cache replacement algorithms

Cache replacement is a well studied field in literature, particularly for web caching. One of the most prominent approaches which is relevant to the replacement strategy adapted in our model is GreedyDual–Size (GDS) [11]. GDS is a cost-aware algorithm that integrates locality information as follows: Cached objects are given a score value based on the cost of bringing them into the cache. When a replacement needs to be made, the content with the lowest cost score $H_{\min}$ gets replaced and the rest of the objects *reduce* their score values by $H_{\min}$. By reducing score values as time goes on and by restoring them only upon a new request, the algorithm manages to seamlessly integrate cost concerns and requests locality in time. Jin and Bestavros [45] analyse temporal locality beyond the properties of content popularity skewness and generalise GDS. The resulted GreedDual$*$ algorithm uses a utility value u in the place of the cost per data unit for fetching an object that is used in GDS, along with an cache-ageing factor L as follows: Objects are assigned with $u + L$ upon cache hits, while L gets the value that was assigned to the most recently evicted object. Hence, L works as an *inflation* factor for cached objects in order to reflect the importance of locality due to temporal requests correlation, versus the importance of long-term content popularity.

Based on our discussion in Section 3.2 on page 14, our model adapts a gain-based replacements approach which has comparable design features with [11] and, especially, [45]. It captures requests locality via Q^l, while the performance evaluation results denote that the aggregated mobility information in Q^l can also adapt well to content popularity. Also, our model extension approximates content popularity f_s^l in a way (see formula (10)) which captures temporal correlation, and further tunes its significance on cache replacement actions based on local demand information.

7. Conclusions and future work

We present a novel Efficient Mobility-based Caching (EMC) distributed model along with content popularity and legacy caching model extensions. Our solution has significant design advantages over other proactive approaches, the most important of which lie in its ability (i) to address *niche* mobile demand, (ii) to dynamically *tune* the contribution of mobile requests' popularity and users' mobility prediction on cache actions, and (iii) to take on-the-fly cache decisions based on contemporary, short timescale local mobility information. By design, our approach targets less popular or personalised content that is unaddressed by other proactive approaches in literature and CDNs, and can be applied to heterogeneous wireless network environments in order to yield monetary and delay cost gains for users with positive implications on QoE. According to credible studies and network forecast reports, such niche content requests represent 20–40% of Internet demand, with mobile video in particular following the growing popularity trend of *personalised* videos published in social networks. As we discuss, even if 60–80% of video demand continuous to account for popular content in the future, still a significant 20–40% will refer to niche videos. To our knowledge, our approach is the most appropriate for complementing CDNs because it is the *only* one that intercepts requests for niche content which is otherwise missed due the providence of CDNs exclusively for popular objects.

Regarding the performance of our solution, among our most notable findings, we show that gains are good from mobility prediction against the cases of applying no, popularity-only or naïve local proactive caching in scenarios that combine different caching demand levels, video catalogues and mobility models, among other system parameter combinations. Cache decisions based only on mobility prediction appear to adapt well to temporal locality due to using short timescale information which allow to capture changes in temporal locality. This outcome largely coincides with the conclusions in [4] which relate prediction accuracy to the latest mobility information in urban environments due to the higher road network complexity, traffic congestion and the variety of mobility habits and routes. Extending our model with cache replacements can (even substantially) improve performance in terms of average gains and robustness against system parameters, yet at the cost of a considerable computational overhead. However, our basic

model already yields an overall good performance which in certain scenarios can be very close to its extended counterparts, especially when including backhaul gains. This observation is important given the added complexity of cache replacements. In addition, we observe that the performance of mobility-based caching appears to improve with the level of popularity skewness, approaching close to the high and robust gains of the extended model with cache replacements. Furthermore, we present scenarios in which the level of caching demand has an important impact on performance, more important than the size of the video catalogue and, finally, unlike other work in literature we try to study the delay cost gains from local proactive caching w.r.t. the presence of CDNs. Our conclusion is that our model can have significantly more robust gains than proactive approaches targeted on popular content.

For future work we intend to investigate video-streaming and hierarchical cache structure applications in heterogeneous wireless networking scenarios, as well as to explore an entirely different spectrum of applications such as assisting name resolution in Information Centric Networks with proactive in-network caching.

Acknowledgement

The work presented in this paper was supported by the European Union funded H2020 ICT project POINT, under contract 643990.

References

[1] Cisco Systems, Cisco visual networking index: global mobile data traffic forecast update, 2016–2020, 2016,
[2] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, Y. Selén, 5G radio access, Ericsson Rev. 6 (2014) 2–7.
[3] V. Chandrasekhar, J. Andrews, A. Gatherer, Femtocell networks: a survey, Commun. Mag. IEEE 46 (9) (2008) 59–67, doi:10.1109/MCOM.2008.4623708.
[4] F. Zhang, C. Xu, Y. Zhang, K. Ramakrishnan, S. Mukherjee, R. Yates, T. Nguyen, EdgeBuffer: caching and prefetching content at the edge in the MobilityFirst future Internet architecture, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a, 2015, pp. 1–9, doi:10.1109/WoWMoM.2015.7158137.
[5] E. Bastug, J.-L. Guenego, M. Debbah, Proactive small cell networks, in: Telecommunications (ICT), 2013 20th International Conference on, 2013, pp. 1–5, doi:10.1109/ICTEL.2013.6632164.
[6] E. Bastug, M. Bennis, M. Debbah, Living on the edge: the role of proactive caching in 5G wireless networks, Commun. Mag. IEEE 52 (8) (2014) 82–89, doi:10.1109/MCOM.2014.6871674.
[7] M.H. Cheung, J. Huang, Optimal delayed Wi-Fi offloading, in: Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on, 2013, pp. 564–571.
[8] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, G. Caire, Femtocaching: wireless video content delivery through distributed caching helpers, in: INFOCOM, 2012 Proceedings IEEE, 2012, pp. 1107–1115, doi:10.1109/INFCOM.2012.6195469.
[9] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, H.M. Levy, On the scale and performance of cooperative web proxy caching, in: ACM SIGOPS Operating Systems Review, vol. 33, ACM, 1999, pp. 16–31, doi:10.1145/319151.319153.
[10] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and Zipf-like distributions: evidence and implications, in: INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 1, 1999, pp. 126–134 vol.1, doi:10.1109/INFCOM.1999.749260.
[11] P. Cao, J. Zhang, K. Beach, Active cache: caching dynamic contents on the web, Distrib. Sys. Eng. 6 (1) (1999) 43–50, doi:10.1088/0967-1846/6/1/305.
[12] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, S. Moon, I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, 2007, pp. 1–14, doi:10.1145/1298306.1298309.
[13] H. Yu, D. Zheng, B.Y. Zhao, W. Zheng, Understanding user behavior in large-scale video-on-demand systems, in: ACM SIGOPS Operating Systems Review, ACM, 2006, pp. 333–344, doi:10.1145/1218063.1217968.
[14] A. Finamore, M. Mellia, M.M. Munafò, R. Torres, S.G. Rao, YouTube everywhere: impact of device and infrastructure synergies on user experience, in: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, ACM, 2011, pp. 345–360, doi:10.1145/2068816.2068849.
[15] J. Xu, J. Liu, B. Li, X. Jia, Caching and prefetching for web content distribution, Comput. Sci. Eng. 6 (4) (2004) 54–59, doi:10.1109/MCSE.2004.5.
[16] D. Beaver, S. Kumar, H.C. Li, J. Sobel, P. Vajgel, et al., Finding a needle in haystack: Facebook's photo storage., in: OSDI, vol. 10, 2010, pp. 1–8.
[17] J.L. García-Dorado, A. Finamore, M. Mellia, M. Meo, M. Munafo, Characterization of ISP traffic: trends, user habits, and access technology impact, Netw. Serv. Manage. IEEE Trans. 9 (2) (2012) 142–155, doi:10.1109/TNSM.2012.022412.110184.
[18] S. Borst, V. Gupta, A. Walid, Distributed caching algorithms for content distribution networks, in: INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–9, doi:10.1109/INFCOM.2010.5461964.
[19] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V.A. Siris, G.C. Polyzos, Caching and mobility support in a publish-subscribe internet architecture, Commun. Mag. IEEE 50 (7) (2012) 52–58, doi:10.1109/MCOM.2012.6231279.
[20] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, J. Wilcox, Information-centric networking: seeing the forest for the trees, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, ACM, 2011, p. 1, doi:10.1145/2070562.2070563.
[21] S. Martello, P. Toth, Knapsack problems: algorithms and computer implementations. 1990, Discr. Math. Optim. (1990).
[22] I.D. Baev, R. Rajaraman, Approximation algorithms for data placement in arbitrary networks, in: Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2001, pp. 661–670.
[23] I. Baev, R. Rajaraman, C. Swamy, Approximation algorithms for data placement problems, SIAM J. Comput. 38 (4) (2008) 1411–1429, doi:10.1137/080715421.
[24] M.R. Korupolu, C.G. Plaxton, R. Rajaraman, Placement algorithms for hierarchical cooperative caching, J. Algo. 38 (1) (2001) 260–302.
[25] K. Katsaros, G. Xylomenos, G. Polyzos, GlobeTraff: a traffic workload generator for the performance evaluation of future Internet architectures, in: New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on, 2012, pp. 1–5, doi:10.1109/NTMS.2012.6208742.
[26] M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser, CRAWDAD dataset epfl/mobility (v. 2009-02-24), 2009, (Downloaded from http://crawdad.org/epfl/mobility/20090224). 10.15783/C7J010
[27] S.T. Kouyoumdjieva, È?lafur Ragnar Helgason, G. Karlsson, CRAWDAD dataset kth/walkers (v. 2014-05-05), 2014, (Downloaded from http://crawdad.org/kth/walkers/20140505). 10.15783/C7Z30C
[28] P.S. Prasad, P. Agrawal, Mobility prediction for wireless network resource management, in: System Theory, 2009. SSST 2009. 41st Southeastern Symposium on, IEEE, 2009, pp. 98–102.
[29] L. Mihaylova, D. Angelova, S. Honary, D.R. Bull, C.N. Canagarajah, B. Ristic, Mobility tracking in cellular networks using particle filtering, Wireless Commun. IEEE Trans. 6 (10) (2007) 3589–3599.
[30] P. Deshpande, A. Kashyap, C. Sung, S.R. Das, Predictive methods for improved vehicular Wi-Fi access, in: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services, ACM, 2009, pp. 263–276, doi:10.1145/1555816.1555843.
[31] V.A. Siris, X. Vasilakos, G.C. Polyzos, A Selective neighbor caching approach for supporting mobility in publish/subscribe networks, in: Proc. of ERCIM Workshop on eMobility. Held in Conjunction with WWIC, 2011, p. 63.
[32] I. Burcea, H.-A. Jacobsen, E. De Lara, V. Muthusamy, M. Petrovic, Disconnected operation in publish/subscribe middleware, in: Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on, IEEE, 2004, pp. 39–50, doi:10.1109/MDM.2004.1263041.
[33] F. Malandrino, M. Kurant, A. Markopoulou, C. Westphal, U.C. Kozat, Proactive seeding for information cascades in cellular networks, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 1719–1727, doi:10.1109/INFCOM.2012.6195543.
[34] V. Gonçalves, N. Walravens, P. Ballon, how about an app store?¥ enablers and constraints in platform strategies for mobile network operators, in: Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), 2010 Ninth International Conference on, IEEE, 2010, pp. 66–73.
[35] E. Bastug, K. Hamidouche, W. Saad, M. Debbah, Centrality-based caching for mobile wireless networks, 1st KuVS Workshop on Anticipatory Networks, 2014.
[36] K. Hamidouche, W. Saad, M. Debbah, Many-to-many matching games for proactive social-caching in wireless small cell networks, in: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on, IEEE, 2014, pp. 569–574, doi:10.1109/WIOPT.2014.6850348.
[37] S. Tamoor-ul Hassan, M. Bennis, P.H. Nardelli, M. Latva-Aho, Caching in wireless small cell networks: A storage-bandwidth tradeoff (2016).
[38] V.A. Siris, D. Kalyvas, Enhancing mobile data offloading with mobility prediction and prefetching, ACM SIGMOBILE Mob. Comput. Commun. Rev. 17 (1) (2013) 22–29, doi:10.1145/2502935.2502940.
[39] V. Siris, X. Vasilakos, D. Dimopoulos, Exploiting mobility prediction for mobility & popularity caching and dash adaptation, A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on (to appear), 2016.
[40] D. Dimopoulos, C. Boursinos, V.A. Siris, Multi-source mobile video streaming: load balancing, fault tolerance, and offloading with prefetching, in: Proceedings of the 9th Intl Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (Tridentcom), 2014, doi:10.1007/978-3-319-13326-3_26.
[41] V.A. Siris, M. Anagnostopoulou, D. Dimopoulos, Improving mobile video streaming with mobility prediction and prefetching in integrated cellular-WiFi networks, in: Mobile and Ubiquitous Systems: Computing, Networking, and Services, Springer, 2014a, pp. 699–704, doi:10.1007/978-3-319-11569-6_56.

[42] V. Siris, X. Vasilakos, G. Polyzos, Efficient proactive caching for supporting seamless mobility, in: A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on, 2014b, pp. 1–6, doi:10.1109/WoWMoM.2014.6918952.

[43] X. Vasilakos, V.A. Siris, G.C. Polyzos, M. Pomonis, Proactive selective neighbor caching for enhancing mobility support in information-centric networks, in: Proceedings of the second edition of the ICN workshop on Information-centric networking, ACM, 2012, pp. 61–66, doi:10.1145/2342488.2342502.

[44] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, K.K. Ramakrishnan, Optimal content placement for a large-scale vod system, in: Proceedings of the 6th International COnference, ACM, 2010, p. 4, doi:10.1145/1921168.1921174.

[45] S. Jin, A. Bestavros, Greedydual web caching algorithm: exploiting the two sources of temporal locality in web request streams, Comput. Commun. 24 (2) (2001) 174–183, doi:10.1016/S0140-3664(00)00312-1.

Xenofon S. Vasilakos is a Ph.D. student at the Athens University of Economics and Business (AUEB) and a member of the Mobile Multimedia Laboratory. He obtained his B.Sc. in Informatics from AUEB in 2007 and his M.Sc. degree in Parallel and Distributed Computer Systems in 2009 from the Vrije Universiteit Amsterdam. His current research interests include Information-Centric Networking architectures, protocols and distributed solutions for the Future Internet, with an emphasis on rendezvous networks and seamless mobility support. He currently participates in the EU funded H2020 ICT project POINT. In the past, he has participated in the EU funded FP7 projects PSIRP and PURSUIT, as well as in the Greek government funded project I-CAN on clean-slate Information-Centric Networking architectures. CV: http://pages.cs.aueb.gr/~xvas/pdfs/CVdetailedXV.pdf

Vasilios A. Siris received a degree in physics from the National and Kapodistrian University of Athens, Greece in 1990, the MS degree in computer science from Northeastern University, Boston in 1992, and the PhD degree in computer science from the University of Crete, Greece in 1998. He is an associate professor in the Department of Informatics, Athens University of Economics and Business, where he is since 2009. From 2002 to 2008, he was an assistant professor at the University of Crete and research associate at the Institute of Computer Science of FORTH (Foundation for Research and Technology – Hellas). In Spring 2001, he was a visiting researcher at the Statistical Laboratory of the University of Cambridge, and in Summer 2001 and 2006, he was a research fellow at the research laboratories of British Telecommunications (BT), United Kingdom. His current research interests include resource management and traffic control in wired and wireless networks, traffic measurement and analysis for monitoring QoS/QoE and, and architectures of mobile communication systems and future networks. He has served as the general chair or technical program chair for various international conferences and workshops, such as IEEE WoWMoM 2009, HotMESH 2011, IEEE Broadband Wireless Access 2011/2012, workshop on Quality, Reliability, and Security in Information-Centric Networking (Q-ICN) 2014, and IEEE PERCOM workshop on Information Quality and Quality of Service for Pervasive Computing (IQ2S) 2015. He is/was the principal investigator or coordinator for many research and development projects funded by the European Commission, the Greek government, the European Space Agency, and the industry.

George C. Polyzos, Professor of Computer Science at AUEB, founded and is leading the Mobile Multimedia Laboratory (MMlab). Previously, he was Professor of Computer Science and Engineering at the University of California, San Diego, where he was co-director of the Computer Systems Laboratory, member of the Steering Committee of the Center for Wireless Communications, and Senior Fellow of the San Diego Supercomputer Center. After joining UCSD he focused his research on Internet based multimedia and wireless communications with emphasis on multimedia dissemination, automatic media adaptation and addressing heterogeneity. More recently, Prof. Polyzos and the MMlab participated in the EU FP7 projects PSIRP and PURSUIT that developed the Information–Centric Networking (ICN) Publish- Subscribe Internet (PSI) architecture and the ESA-funded project ϕSAT, which investigated "The Role of Satellite in Future Internet Services." He co-authored a comprehensive survey article on ICN. Prof. Polyzos was also an organizer of the EIFFEL Think Tank, on the Steering Board of the Euro–NF Network of Excellence and head of its "Socio–Economic Aspects" and "Trust, Privacy and Security" joint research activities. Dr. Polyzos received his Diploma in EE from the National Technical University, Athens, Greece and his M.A.sc. in Electrical Engineering and Ph.D. in Computer Science from the University of Toronto. He has been reviewer or panelist for many research funding agencies, including the European Commission, the US NSF, the California MICRO program, the Swiss NSF, the European ERA-Net, and the Greek GSRT; he has also been on the editorial board and guest editor for scientific journals, on the program committees of many conferences and workshops, the chair of the Steering Committee of the ACM SIGCOMM conference on Information-Centric Networking and on the Steering Committee of the Wireless and Mobile Networking Conference, WG 6.8, IFIP TC6. His current research interests include Internet architecture and protocols, ICN, mobile multimedia communications, ubiquitous computing, security, privacy, wireless networks, and performance evaluation of computer and communications systems. CV: http://niovi.aueb.gr/~gcp/CV-EN.pdf