Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud

Luigi Vigneri, Thrasyvoulos Spyropoulos, Chadi Barakat

Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud

Luigi Vigneri, *Member, IEEE,* Thrasyvoulos Spyropoulos, *Member, IEEE,*
and Chadi Barakat, *Senior Member, IEEE*

***Abstract*—Densification through small cells and caching in base stations have been proposed to deal with the increasing demand for Internet content and the related overload on the cellular infrastructure. However, these solutions are expensive to install and maintain. Instead, using vehicles acting as mobile caches might represent an interesting alternative. In our work, we assume that users can query nearby vehicles for some time, and be redirected to the cellular infrastructure when the deadline expires. Beyond reducing costs, in such an architecture, through vehicle mobility, a user sees a much larger variety of locally accessible content within only few minutes. Unlike most of the related works on delay tolerant access, we consider the impact on the user experience by assigning different retrieval deadlines per content. In our paper, we provide the following contributions: (i) we model analytically such a scenario; (ii) we formulate an optimization problem to maximize the traffic offloaded while ensuring user experience guarantees; (iii) we propose two variable deadline policies; (iv) we perform realistic trace-based simulations, and we show that, even with low technology penetration rate, more than 60% of the total traffic can be offloaded which is around 20% larger compared to existing allocation policies.**

I. Introduction

THE widespread availability of handheld devices, such as smartphones or tablets, and the content-centric nature of Web 2.0 applications are driving a massive increase in the mobile traffic demand [1]. While operators are slowly upgrading their networks to LTE and considering a number of communication technologies for beyond 4G networks (e.g., massive MIMO, Collaborative Multi-Point), such upgrades are quite expensive, and might not even be able to keep up with the demand. As an alternative, operators are turning towards densification through small cells (SCs) which promises improved spectral efficiency at smaller capital/operational expenditures (CAPEX/OPEX). However, introducing a large number of SCs requires significant upgrades to the backhaul network which is predicted to become the new bottleneck [2].

Caching popular content in SCs has been proposed as a solution to alleviate the congestion of the backhaul and core networks [3]–[5], but comes with its own drawbacks: (i) extensive SC coverage is necessary to ensure enough traffic is offloaded from the macro-cells, but the initial experience suggests larger CAPEX/OPEX per site than initially predicted [6]; (ii) while caching popular content deep inside the operator's core network promises good hit rates [7], initial studies based on real data are pessimistic [8]. Recently, instead of (or in addition to) fixed operator-installed SCs, it has been proposed the use of private or public transportation as storage points and mobile relays to store popular content [9]. We refer to such storage cloud as *vehicular cloud* which is the common terminology used in related work [10], also motivated by the fact that computing capabilities for vehicles are expected in the future. These mobile caches can be controlled by mobile network operators (MNOs) through a cellular interface. In urban environments, the number of vehicles is expected to be considerably higher than in any envisioned SC deployment. Hence, the sheer number of vehicles along with the lower CAPEX/OPEX involved makes this an interesting alternative.

In this paper, we exploit such a vehicular cloud to store popular content (e.g., software updates, not real-time videos) in order to offload part of the mobile traffic demand. We build a model where a user requesting a content queries nearby vehicles and, if the content is not found, is redirected to the main cellular infrastructure. However, since caches will be quite small compared to the daily catalogue of content, the user might not be within range of any cache storing the requested content at that time. To alleviate this, we propose that each request can be delayed for a small amount of time, if there is a local cache miss. While the idea of delay tolerance has already been extensively discussed in literature, in this work we introduce three fundamental novelties:

Vehicle storage capacity "virtually" extended. Delayed offloading to small cells (with and without local storage) has already been considered (e.g., via WiFi access points [11], [12]). However, most of these works *require the user to move* in order to encounter new base stations and see new caches. This is problematic as most users exhibit a nomadic behavior, staying in the same location for long periods. As a result, it has been consistently reported that such delayed offloading architectures require TTLs (Time to Live) in the order of half to a couple of hours to demonstrate performance benefits [11]–[13]. Instead, when caches are on vehicles, especially in a dense urban environment, a static or slowly moving user will see a much larger number of caches within the same amount of time, thus *virtually extending the size of the accessible local storage*. This leads to better hit rates with considerably smaller deadlines (in the order of a few minutes, see Section VI).

Variable deadlines. The majority of edge caching related works are operator-centric, aiming at policies that exclusively minimize the load on the cellular infrastructure. In most delayed offloading settings, the worst-case delay TTL guarantee offered to the user is usually *fixed* for all content requests and set to large values in order to offload a considerable amount of traffic, as explained earlier. Conversely, in this work we allow the operator to set different deadlines for different content. This variability in the TTL brings two advantages: first, it allows to increase the percentage of the traffic offloaded as we

will see in the rest of the paper; second, these deadlines can be adapted according to the specific characteristics of the content (e.g., size) in order to improve user Quality of Experience (QoE) as we explain below.

User QoE-aware offloading. QoE is a measure of the delight or annoyance of a user's experience with a service (e.g., phone call, streaming). While other models can be possible, we choose to evaluate the user QoE according to the experienced *slowdown* which has become popular in recent queuing theory literature [14]. This metric relates the waiting delay with the "net" download time. For example, a user requesting a web page of a few MBs (normally taking some seconds) will be quite frustrated if she has to wait an extra 1-2 minutes to encounter a vehicle caching that web page. However, a user downloading a large video or software file might not even notice an extra 1-2 minutes delay. Specifically, in our framework an MNO can calibrate the user experience by setting a required slowdown which upper bounds the tail behavior of the response time. Unlike related works that use large TTLs, tuning the waiting time per content ensures maximum offloading with little QoE degradation.

While there are a number of additional architectural and incentive-related questions to consider (see Section VII), the main focus of this paper is on the modelling of the above scenario and on the formulation of a corresponding (nontrivial) optimization problem. We provide more details about related work and respective novelty in the next section. The main contributions of the paper can be summarized as follows:

1) We model the problem of maximizing the percentage of traffic offloaded through the vehicular cloud considering the user QoE and a large range of realistic conditions.
2) We solve the problem in 1) presenting two variable deadline caching policies: QoE-Aware Caching (QAC) which introduces an approximation on the generic formulation; QoE-Aware Caching for Small Content (QAC-SC) which provides better offloading gains for content of small sizes.
3) We validate our findings using simulations with real traces for vehicle mobility and content popularity. We show that our system can offload a considerable amount of bytes with modest technology penetration ($< 1\%$ of vehicles participating in the cloud) and low mean slowdown (that leads to average deadlines of a few minutes).
4) We study the impact of different user QoE guarantees on operator- and user-related performance, and compare our QAC and QAC-SC with some fixed deadline policies.

The rest of the paper is organized as follows: in Section II we compare our work with the previous literature; in Section III, we define the system model with the main assumptions; then, in Section IV, we present the mathematical formulation of the problem, and we solve a reasonable approximation (since the original problem is hard); in Section V we introduce two policies specific for small content; we validate our results through real-trace based simulations in Section VI; finally, we discuss about architectural details and incentives in Section VII, and conclude our paper with a summary and future work in Section VIII.

II. Related work

The rapid increase in the mobile traffic demand has led to a large number of proposals to mitigate the load on the cellular infrastructure. The ones most closely related to our approach can be roughly categorized as follows.

A. Caching at the edge

Caching at the edge of the network has been deeply investigated by researchers lately. In this context, traditional solutions concern adding storage capacity to SCs [4], [5] and/or to some intermediate nodes within the network [15]. Golrezaei *et al.* [4] propose to replace backhaul capacity with storage capacity at the SC access points (APs), called *helpers*; the challenge faced by the authors was in the analysis of the optimum way of assigning content to the helpers in order to minimize the expected download time. Poularakis *et al.* [5] focus their attention on video requests trying to optimize the service cost and the delivery delay; in their framework, pre-stored video files can be encoded with two different schemes in various qualities. Finally, Zhou *et al.* [15] introduce an information-centric heterogeneous network framework aiming at enabling content caching and computing; due to the virtualization of the whole system, communication, computing, and caching resources can be shared among all users associated with different virtual service providers.

While such distributed caching schemes for SCs provide very interesting theoretical insights and algorithms, they face some key shortcomings. A large number of SCs is required for an extensive enough coverage, which comes at a high cost [6]. E.g., in a macro-cell of a radius of a few kilometers, it is envisioned to place 3-5 SCs, of range a few hundred meters. By contrast, in an urban environment, the same area will probably contain thousands of vehicles. Furthermore, the smaller size of edge caches and smaller number of users per cell, raises the question whether enough overlap in user demand would be generated locally to have a high enough hit ratio, when real traffic is considered. Delayed content access is supposed to overcome such limitation as explained next. Another key difference is that SCs are static and user locations are supposed to be known. In our approach, we actually allow more generic mobility patterns (e.g., no assumptions about user locations) and also introduce delayed access which "mixes" SCs and users.

B. Delayed content access

To alleviate the aforementioned problem of requests overlap at a low cost, a number of works introduce delayed access. This can be seen as an enforced delay until a WiFi access point is encountered to offload the cellular connection to a less loaded Radio Access Technology (RAT) [11]–[13], or until to reach peer nodes in a P2P infrastructure [16]. For example, Balasubramanian *et al.* [11] develop a system to augment mobile 3G capacity with WiFi, using two key ideas: delay tolerance and fast switching. This enforced delay virtually extends the coverage of WiFi APs, allowing a larger ratio of connections to be offloaded than the mere physical coverage of WiFi APs allows. In other works [17], a different deadline

is assigned to each content. However, these deadlines are problem input parameters and cannot be used to improve performance (e.g., the amount of data offloaded, QoE), as we do in our paper. Nevertheless, as explained earlier, these approaches *require the user to move* in order to encounter new base stations and new caches. User mobility is often nomadic and slow, requiring the respective algorithms to enforce very large content access delays (often in the order of hours), before any performance improvement is perceived by the operator. Instead, in our paper we present two main novelties: (i) having the SC and cache move, naturally happening when placed on vehicles, the operator can offload up to 60% of its traffic with minimum QoE impact; (ii) while other works consider pre-assigned deadlines, we allow variable delay tolerance per content, and also allow the operator to optimize it (by setting an upper limit on the slowdown) in order to improve QoE or maximize the offloaded data.

C. Caching on mobile devices

Apart from small cells, researchers have also been proposing to use mobile devices to offload content through opportunistic communications [3], [18]. Bao *et al.* [3] exploit the possibility of serving user requests from other mobile devices located geographically close to the user; the goal of the work is to explore a practical way of offloading cellular traffic via D2D communications, exploiting the observation that cellular networks are strained when many people located in a small area request for content (e.g., concerts, stadiums). In the work of Han *et al.* [19], mobile devices store content and propagate the information opportunistically; the challenge is to find a set of users where to offload the information. In addition, Li *et al.* [18] also takes into account the analysis of social behavior and preference of mobile users. Nevertheless, having mobile devices in tethering mode, storing even a small subset of the total content catalogue and having them to serve constantly incoming requests from other users seem to put an unrealistically high toll on the already limited battery, storage and processing resources of handheld devices. On the other hand, placing and powering up a large hard disk and a simple AP somewhere inside the vehicle seems to pose much fewer challenges for modern cars. To sum up, compared to user equipments acting as relays and caches, the vehicular cloud offers considerably more storage and processing power, thus lowering the adoption barrier significantly.

D. Vehicles as cellular infrastructure helpers

Recently, a few works have suggested to exploit vehicular networks to store content [20]–[22]. Zhang *et al.* [20] propose a P2P scheme to improve the performance of content sharing in intermittently connected vehicular networks. Zhao *et al.* [21] adopt the idea of carry and forward content where a moving vehicle carries information until a new vehicle moves into its vicinity: the authors make use of the predictable vehicle mobility to reduce the content delivery delay. Nevertheless, the majority of these works do not consider a common cloud maintained by the mobile network operator. Conversely, in a previous work, we have introduced the idea of vehicular cloud used to offload part of the traffic and accessible by handheld devices [22]. However, the paper only mentions initial thoughts about the architecture, without dealing with QoE or variable deadlines. The hype around vehicular networks as part of the cellular infrastructure has been confirmed by the launch of new companies like Veniam which offers network connectivity to public transportation, and is rapidly spreading around the world [9].

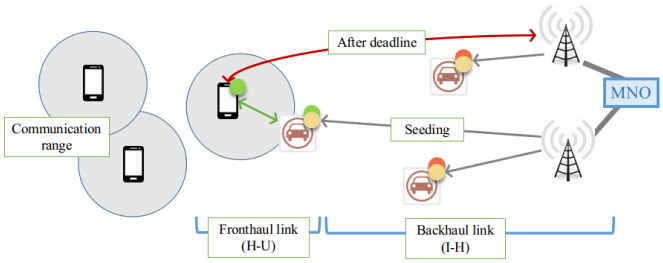


Fig. 1: An MNO pushes content in vehicles according to the chosen policy; when a mobile device is in the communication range of a vehicle, some of the requested chunks are offloaded; if the download is not finished within the deadline, the device is redirected through the cellular infrastructure.

III. System model

In this section we introduce the system model with the related assumptions that will be used to formulate an optimization problem maximizing the traffic offloaded through the vehicular cloud while accounting for the end-user QoE.

A. Content access protocol

We consider a network with three types of nodes:

- *Infrastructure nodes* ($\mathcal{I}$): base stations or macro-cells; their role is to seed content into vehicles and to serve user requests when the deadline expires.
- *Helper nodes* ($\mathcal{H}$): vehicles such as cars, buses, taxis, trucks, etc., where $|\mathcal{H}| = h$; these are used to store popular content and to serve user requests at low cost through a direct vehicle to mobile node link.
- *End-user nodes* ($\mathcal{U}$): mobile devices such as smartphones, tablets or netbooks; these nodes request content to $\mathcal{H}$ and $\mathcal{I}$ nodes (the last ones are only contacted when the deadline expires and the content is still not entirely downloaded).

The basic protocol is made up of three phases (Fig. 1):

- *Backhaul link* ($\mathcal{I} \rightarrow \mathcal{H}$): $\mathcal{I}$ nodes place content in $\mathcal{H}$ nodes according to the chosen allocation policy. These policies are the main outcome of this paper. We refer to this phase as *seeding*. This phase is repeated at the beginning of operator selected time windows to adjust to varying content access patterns. If seeding is performed during off-peak times, its cost can be considered equal to 0. In our work, without loss of generality, we will focus on this scenario[1]. Vehicles could be treated as end devices, in which case this becomes a *normal radio access link*.

[1]The generic case (i.e., non-null seeding cost) is a straightforward extension when seeding time windows are large enough to amortize content seeding.

- *Fronthaul link* ($\mathcal{H} \rightarrow \mathcal{U}$): an end user node can request content i to the vehicles that are inside her communication range[2]. If content i is found, then the $\mathcal{U}$ node can download bytes from the vehicle during the contact. If the download is not terminated, then the requesting mobile user will query nearby vehicles for a time equal to y_i. This deadline is decided for that content i by the MNO according to the allocation policy chosen[3]. The related local access cost is assumed to be 0. This link could use the operator's main RAT or a different RAT like WiFi, and be controlled by the operator (so subject to central scheduling). Alternatively, the devices themselves could communicate directly (e.g., using WiFi direct). Depending on these choices, the initial content request could be made directly to nearby $\mathcal{H}$ or $\mathcal{I}$ nodes that can decide to redirect the user to the cloud.
- ($\mathcal{I} \rightarrow \mathcal{U}$): in case of a content not successfully downloaded within y_i, the $\mathcal{U}$ node's request will be served (partially or entirely) by the cellular infrastructure. The cost to get content i from $\mathcal{I}$ is equal to the number of bytes downloaded from the cellular infrastructure.

As a final note, in some settings $\mathcal{H}$ nodes could also communicate with each other, e.g., to update their local caches or to fetch a content over multiple hops, in case of a local miss. Nevertheless, such multi-hop (MANET-type or DTN-type) schemes might considerably increase the complexity of our approach, while only bringing incremental benefits. Further details about the architecture are provided in Section VII.

B. Main assumptions

A.1 - Catalogue. Let K be the set of all possible contents that users might request (also defined as "catalogue"), where $|K| = k$. Let further c be the size of the cache in each vehicle. We make the natural assumption that $c \ll k$. A content $i \in K$ is of size s_i (in MB) and is characterized by a popularity value ϕ_i measured as the expected number of requests within a seeding time window from all users and all cells. Similar to a number of works on edge caching [5], [23], we assume this time window to be a system parameter chosen by the MNO. Every time window, the MNO refreshes its caches installed in vehicles according to the new estimated popularity. However, while it is reasonable to assume the content size is known, predicting the popularity of a content is more challenging. Nevertheless, several studies have confirmed that simple statistical models (e.g., ARMA) along with content type characteristics can help to have good estimation of the number of requests, at least in the immediate future [24].

A.2 - Mobility model. We assume that the inter-meeting times T_{ij} between a user requesting content $i \in K$ and a vehicle $j \in H$ are IID random variables characterized by a known cumulative distribution function (CDF) $F_T(t) = \mathbf{P}[T_{ij} \leq t]$ with mean rate λ. Let further T_i be the inter-meeting times between a user requesting content $i \in K$ and *any* vehicle storing such a content. This model does not make any assumption on the individual user and vehicle mobility patterns and can capture a number of inter-contact time models proposed in related literature such as exponential, Pareto, or mixed models.

A.3 - Cache model. Let $x_{ij} \in \{0, 1\}$, $i \in K, j \in \mathcal{H}$ be an indicator variable denoting if helper node j stores content i. Hence, we assume $\mathcal{H}$ nodes to *store the whole content*, i.e., fractional storage is not allowed. Let further $x_i = \sum_{j \in \mathcal{H}} x_{ij}$ denote the number of $\mathcal{H}$ nodes storing content i. The vector $\mathbf{x} = \{x_i\}_{i \in K}$ will be the control variable for our optimal cache allocation problem. Note that given the assumption of IID mobility, it suffices to optimize the total number of copies x_i without considering the per vehicle variables x_{ij} any more.

A.4 - Chunk download. Opportunistic meetings between $\mathcal{U}$ and $\mathcal{H}$ nodes are described by the well-known "protocol model" that uses a simplified description of the physical layer: two nodes communicate if their physical distance is smaller than some collaborative distance determined by the power level for each transmission. We refer to such meetings as *contacts*. Let b_{ij} be the number of bytes downloaded from content i by a $\mathcal{U}$ node during the j^{th} meeting. b_{ij} are positive IID continuous random variables having equal mean μ and variance σ^2. Let further M_i be a point process counting the number of contacts within y_i. Then, we define $B_i \triangleq \sum_{j=1}^{M_i} b_{ij}$ as the number of bytes downloaded within y_i from content i.

A.5 - QoE metric. First, we define $t_i \triangleq s_i/r$ as the *net* download time of content i by a user, i.e., the amount of time it takes to download the content (excluding any potential waiting time to encounter vehicles holding the content), where r is the download rate from the cellular infrastructure. As for videos, t_i can be thought of as the video duration (and r as the playout rate). Then, we introduce the *maximum slowdown per content* that ties content download time to its size as $\omega_i \triangleq \frac{y_i + t_i}{t_i} = 1 + \frac{y_i}{s_i/r}$. This represents the *maximum slowdown* imposed by our system, when the content is fetched from the infrastructure. The larger ω_i is, the worse the impact of the allocation policy on user experience. This is in fact a *worst case* metric, because if the content is downloaded before the deadline expires, say at some time $d_i < y_i$ (i.e., there is a cache hit), the real slowdown is lower and equal to $1 + \frac{d_i}{t_i}$. Nevertheless, we choose to use the maximum slowdown in our theoretical framework as a more conservative approach for the user, and keep analysis simpler. Furthermore, since the operator's goal is to consider the global QoE (and not only per request), we consider a weighted average of the maximum slowdown according to the content popularity defined as:

$$\Omega(\mathbf{y}) = \frac{1}{\sum_{i=1}^{k} \phi_i} \cdot \sum_{i=1}^{k} \phi_i \cdot \omega_i.$$

For simplicity, we will refer to $\Omega(\mathbf{y})$ as *mean slowdown*. An MNO can use this metric to calibrate the global user QoE of the system by setting a parameter $\omega_{max} > 1$ to upper bound the mean slowdown. This value can be seen as a sort of "budget" available to the MNO that can be reallocated between contents.

[2]The communication range depends on the physical layer technology used between $\mathcal{U}$ and $\mathcal{H}$ nodes.

[3]In reality, deadlines might be application-dependent. This can be easily included in our framework by considering an individual maximum TTL per content (depending on the application). As extreme case, preassigned TTLs have already been discussed in related work [17]. TTLs could also be affected by different types of users (e.g., roaming users might be willing to wait more to get a content at lower cost). In this work, we only consider an average delay-tolerance (which can be tuned by the MNO through the slowdown metric) and we defer further study in this direction to future work.

TABLE I: Notation used in the paper.

Control variables	
x_i	Number of replicas stored for content i across vehicles
y_i	Deadline for content i
Content	
k	Number of content in the catalogue
ϕ_i	Number of requests for content i
s_i	Size of content i
c	Buffer size per vehicle
Mobility	
T_{ij}	Inter-meeting time between $\mathcal{U}$ and $\mathcal{H}$ nodes
λ	Mean inter-meeting rate with vehicles
M_i	Number of contacts within y_i
h	Number of vehicles
Chunk download	
b_{ij}	Bytes downloaded per contact
μ, σ^2	Mean and variance of b_{ij}
B_i	Total bytes downloaded for content i
f_{B_i}	Probability density function of B_i
F_{B_i}	Cumulative density function of B_i
QoE parameters	
r	Download rate from cellular network (or video playout rate)
Ω	Mean slowdown
y_{max}	Maximum deadline
ω_{max}	Upper bound on the mean slowdown

Moreover, it can set a maximum tolerable deadline y_{max} to avoid excessively large TTLs for specific content.

We summarize the notation used in the paper in Table I.

IV. Optimal content allocation

Based on the previous system model, we formulate an optimization problem to reduce the load on the cellular infrastructure. In Section IV-A, we show that this problem is complex as it requires the knowledge of B_i. Then, in Section IV-B, we solve the optimization problem under an approximation for content of generic size.

A. Optimization problem

We formulate an optimization problem based on the following ideas: an ideal content allocation should replicate content with higher popularity in many different vehicles in order to increase the probability to find it from a requesting user. Trivially, more replicas lead to smaller waiting times. However, if the *marginal* gain from extra replicas is nonlinear, it might be better to also have some less popular content at the edge. As the storage capacity of each vehicle is limited, our objective is thus to find the optimal replication factor per content to minimize the total load on the cellular infrastructure while accounting for end users QoE:

Problem 1. *The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud:*

$$\underset{\mathbf{x}\in X^k,\mathbf{y}\in Y^k}{\textit{maximize}} \quad \Phi(\mathbf{x},\mathbf{y}) \triangleq \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}\left[\min\{B_i(\mathbf{x},\mathbf{y}), s_i\}\right], \tag{1}$$

$$\textit{subject to} \quad \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h, \tag{2}$$

$$\Omega(\mathbf{y}) \leq \omega_{max}, \tag{3}$$

where $X \triangleq \{a \in \mathbb{N} \mid 0 \leq a \leq h\}$ and $Y \triangleq \{b \in \mathbb{R} \mid 0 \leq b \leq y_{max}\}$ are the feasible regions for the control variables $\{\mathbf{x}, \mathbf{y}\}$.

For each request, the number of bytes offloaded through the vehicular cloud is either equal to s_i, if the content is entirely downloaded from vehicles, or to B_i, otherwise. For popular content, we can consider the expected value of this quantity since the envisioned number of requests during a seeding time window is large (Eq. (1)). The optimization problem is completed by two inequality constraints: Eq. (2) is a constraint on the total capacity, and Eq. (3) on the mean slowdown.

Lemma 4.1. *The following equivalence holds:*

$$\Phi(\mathbf{x},\mathbf{y}) \equiv \sum_{i=1}^{k} \phi_i \cdot \int_0^{s_i} \left(1 - F_{B_i}(t)\right)\ dt,$$

where F_{B_i} is the CDF of B_i.

Proof. The objective function of Eq. (1) can be rewritten as

$$\begin{aligned}\Phi(\mathbf{x},\mathbf{y}) &= \sum_{i=1}^{k} \phi \cdot \mathbf{E}\left[\min\{B_i, s_i\}\right] \\ &= \sum_{i=1}^{k} \phi_i \cdot \left(\int_0^{s_i} t \cdot f_{B_i}(t)\ dt + \int_{s_i}^{+\infty} s_i \cdot f_{B_i}(t)\ dt\right),\end{aligned}$$

where f_{B_i} is the pdf of B_i. The first integral becomes equal to $s_i \cdot F_{B_i}(s_i) - \int_0^{s_i} F_{B_i}(t)dt$ by integration by parts, while the second integral is trivially equal to $s_i \cdot (1 - F_{B_i}(s_i))$. After simplifying the null terms, we obtain Eq. (1). □

Solving Problem (1) requires the knowledge of F_{B_i} and, therefore, of B_i. We prove that the following proposition holds:

Proposition 4.2. *Assume the number of vehicles participating in the vehicular cloud to be large and the mean inter-meeting rate with such vehicles to be small. Then, B_i can be approximated by a compound Poisson process.*

Proof. Let $\{T_{ij}(t), t > 0, j \in \mathcal{H} \text{ s.t. } x_{ij} = 1\}$ be x_i identical and independent renewal processes corresponding to the inter-contact times with vehicles storing content i. The CDF of T_{ij} is $F_T(t)$ with mean λ (see Assumption A.2). Let further $\{T_i(t), t > 0\}$ be the superposition of these processes. According to the Palm-Kintchine theorem [25], $\{T_i(t)\}$ approaches a Poisson process with rate $\lambda \cdot x_i$ if x_i is large[4] and λ is small. A Poisson process can be defined as a counting process that represents the total number of occurrences up to time t. Thus, the total number of contacts within the deadline $M_i = \{T_i(y_i)\}$ is again a Poisson process.

Remember that $B_i \triangleq \sum_{j=1}^{M_i} b_{ij}$. Observe that the reward (bytes downloaded) in each contact is independent of the inter-contact times, i.e., M_i and b_{ij} are independent, and b_{ij} are IID random variables with same distribution. Since M_i is a Poisson process, then B_i is a compound Poisson process. □

Corollary 4.3. *The following statements can be derived from the previous proposition:*

[4]While this assumption (i.e., x_i large) might not always be true, exponential inter-meeting times have been largely used in literature and considered as a good approximation, especially in the tail of the distribution.

1) The first two moments of B_i are given by:

$$\mathbf{E}[B_i] = \mu \cdot \lambda \cdot x_i \cdot y_i,$$
$$Var[B_i] = (\mu^2 + \sigma^2) \cdot \lambda \cdot x_i \cdot y_i.$$

2) The CDF of B_i is given by:

$$F_{B_i}(s_i) = 1 - \mathcal{L}^{-1}\left\{ e^{(b^*_{ij}(s)-1)\cdot\lambda\cdot x_i\cdot y_i}/s \right\}(s_i), \tag{4}$$

where $b^*_{ij}(s)$ is the Laplace transform of b_{ij}.

Proof. 1) Using conditional expectation, the expected value of a compound Poisson process corresponds to:

$$\mathbf{E}[B_i] = \mathbf{E}\left[\sum_{j=1}^{M_i} b_j\right] = \mathbf{E}\left[\sum_{i=1}^{M_i} \mu\right] = \mathbf{E}[M_i \cdot \mu] = \mathbf{E}[M_i] \cdot \mu,$$

where the expectation is calculated using the Wald's equation. It is easy to see that $\mathbf{E}[M_i] = \lambda \cdot x_i \cdot y_i$. Similarly, it is possible to compute the moment of second order of B_i, and then its variance using the total law of variance.

2) A *random* sum of IID random variables has a Laplace transform that is related to the transform of the summed random variables and of the number of terms in the sum, i.e., $B^*_i(s) = M^*_i(b^*_{ij}(s))$, where B^*_i (resp. b^*_{ij}) is the Laplace transform of B_i (resp. b_{ij}) and M^*_i is the $\mathcal{Z}$-transform of M_i. Since the number of meetings within y_i is Poisson distributed (see proof of Theorem 4.2), we can write $B^*_i(s)$ as follows:

$$B^*_i(s) = e^{(b^*_{ij}(s)-1)\cdot\lambda\cdot x_i\cdot y_i}.$$

Moreover, it is well known that the CDF of a continuous random variable X is given by $F_X(x) = \mathcal{L}^{-1}\left\{\frac{\mathcal{L}\{f_X\}}{s}\right\}(s_i)$ where $\mathcal{L}^{-1}\{F(s)\}(t)$ is the inverse Laplace transform of $F(s)$. Thus, $F_{B_i}(s_i)$ corresponds to Eq. (4). □

Corollary 4.4. *Assume M_i to be large. Then, the probability density function of B_i can be approximated by a normal distribution.*

Proof. In principle, the distribution of B_i is hard to determine. However, since in urban environments the number of contacts is expected to be considerably large, B_i can be approximated by a normal distribution [26]. □

All the quantities needed to solve the optimization problem are known from Corollary 4.3, and can be plugged in Eq. (1). However, due to the large number of contents to consider, the related maximization problem cannot be solved efficiently. For this reason, further insights, approximations and specific scenarios will be discussed in the rest of the paper.

B. QoE-Aware Caching (QAC)

Problem (1) is a mixed-integer nonlinear programming (MINLP) problem. MINLP refers to optimization problems with continuous and discrete variables and nonlinear functions in the objective function and/or the constraints, i.e., it includes both nonlinear programming (NLP) and mixed-integer linear programming (MILP) as subproblems.

Proposition 4.5. *Problem* (1) *is an NP-hard combinatorial problem.*

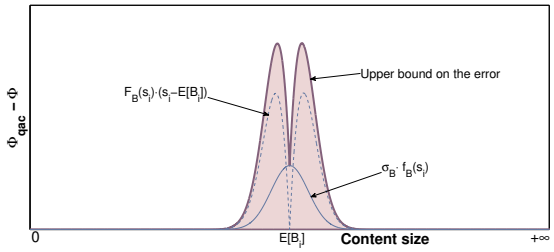


Fig. 2: Error introduced by Φ_{qac} for a fixed value of $\mathbf{E}[B_i]$.

Proof. The problem is NP-hard since it includes MILP as a subproblem [27]. □

What is more, this problem is in general non-convex. This means that the solution can be computed by global optimization methods, but this is generally not an efficient solution as it does not scale to a large number of contents. Similarly to a number of works we consider the *continuous relaxation* of a MINLP which is identical to the mixed-integer problem without the restriction that some variables must be integer. The continuous relaxation brings two fundamental advantages: (i) it is possible to evaluate the quality of a feasible set of solutions; (ii) it is much faster to optimize than the integer problem.

In order to improve tractability of Problem (1), we convert it in a convex problem through an approximation of its objective:

$$\Phi_{qac}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} \phi_i \cdot \min\{\mathbf{E}[B_i], s_i\}. \tag{5}$$

Lemma 4.6. *Let $e \triangleq \Phi_{qac} - \Phi$ be the error introduced by Eq.* (5)*. The following statements hold:*

1) For a given $\mathbf{E}[B_i]$, as the content size s_i tends to 0 or becomes large, the approximation becomes exact, i.e., $\lim_{s_i\to 0} e = \lim_{s_i\to+\infty} e = 0$.

2) The error e is equal to

$$e = \sum_{i=1}^{k} \phi_i \cdot \left[\alpha(s_i) \cdot |s_i - \mathbf{E}[B_i]| + \sigma_{B_i} \cdot f_{B_i}(s_i)\right],$$

where $\alpha(s_i) = \min\{F_{B_i}(s_i), 1 - F_{B_i}(s_i)\}$.

Proof. The proof can be found in Section IX. □

A qualitative analysis of e can be found in Fig. 2 where we can see that the error is concentrated in the region where $s_i \approx \mathbf{E}[B_i]$, and it tends to 0 otherwise. Using the above approximation, Problem (1) can be converted in a *convex* problem that can be solved extremely efficiently and reliably:

Theorem 4.7. *Consider the approximation introduced by Eq.* (5)*. Then, the solution to the following convex problem maximizes the bytes offloaded through the vehicular cloud:*

$$\begin{aligned}
&\underset{\tilde{\mathbf{x}}\in\tilde{X}^k, \tilde{\mathbf{y}}\in\tilde{Y}^k}{\text{maximize}} && \log\left(\sum_{i=1}^{k} \phi_i \cdot e^{\tilde{x}_i+\tilde{y}_i}\right), \\
&\text{subject to} && \tilde{x}_i + \tilde{y}_i \le \log\left(\frac{s_i}{\mu\cdot\lambda}\right), \quad \forall i \in K, \\
& && \sum_i s_i \cdot e^{\tilde{x}_i} \le c \cdot h, \\
& && \Omega(\tilde{\mathbf{y}}) \le \omega_{max},
\end{aligned}$$

where $\tilde{x_i} \triangleq \log x_i$, $\tilde{y_i} \triangleq \log y_i$, $\tilde{X} \triangleq \{a \in \mathbb{R} \mid -\infty \leq a \leq \log h\}$, $\tilde{Y} \triangleq \{b \in \mathbb{R} \mid -\infty \leq b \leq \log y_{max}\}$.

Proof. We rewrite the objective function $\Phi_{qac}(\cdot)$ in an equivalent form that removes the min function:

$$\Phi_{qac}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} \phi_i \cdot \min\{\mathbf{E}[B_i], s_i\}$$
$$= \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}[B_i], \quad \text{s. t.} \quad \mathbf{E}[B_i] \leq s_i, \ \forall i \in K, \quad (6)$$

where the equivalence is true since the related maximization problem will choose the control variables $\mathbf{x}$ and $\mathbf{y}$ such that $0 \leq \mathbf{E}[B_i] \leq s_i$, as any scenario where $\mathbf{E}[B_i] > s_i$ is suboptimal. Remember that $\mathbf{E}[B_i] = \mu \cdot \lambda \cdot x_i \cdot y_i$ from Corollary 4.3. According to Eq. (6), Lemma 1 becomes:

$$\begin{aligned} \underset{\mathbf{x} \in X^k, \mathbf{y} \in Y^k}{\text{maximize}} \quad & \sum_{i=1}^{k} \phi_i \cdot x_i \cdot y_i, \\ \text{subject to} \quad & x_i \cdot y_i \leq \frac{s_i}{\mu \cdot \lambda}, \quad \forall i \in K, \\ & \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h, \\ & \Omega(\mathbf{y}) \leq \omega_{max}. \end{aligned}$$

The above optimization problem is a *geometric program* (GP). A GP is an optimization problem where the objective is a posynomial function and the constraints are posynomial or monomial functions. The main trick to solve a GP efficiently is to convert it to a nonlinear but *convex* optimization problem, since efficient solution methods for general convex optimization problem are well developed [28]. The conversion of a GP to a convex problem is based on a logarithmic change of variables and on a logarithmic transformation of the objective and constraint functions. We apply the following transformations to the above optimization problem:

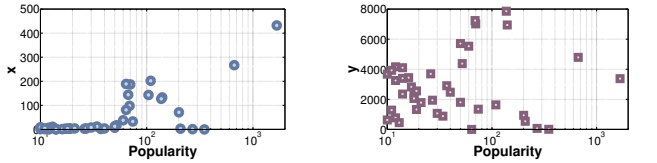
$$\tilde{x_i} \triangleq \log x_i \quad \Leftrightarrow \quad e^{\tilde{x_i}} \triangleq x_i; \qquad \tilde{y_i} \triangleq \log y_i \quad \Leftrightarrow \quad e^{\tilde{y_i}} \triangleq y_i.$$

We obtain a problem expressed in terms of the new variables $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$. By taking the logarithm of objective function and constraints, the related problem becomes convex [28]. □

While this problem seems more complicated in its formulation, NLP is far trickier and always involves some compromise such as accepting a local instead of a global solution. Conversely, a GP can actually be solved efficiently with any nonlinear solver (e.g., MATLAB, SNOPT) or with common optimizers for GP (e.g., MOSEK, GPPOSY). Finally, we use *randomized rounding* [29] on the content allocation which is a widely used approach for designing and analyzing such approximation algorithms. We expect the rounding error to be low since the number of copies per content is usually large (then the decision whether rounding up or down has only a marginal effect in the objective function). To validate this, in Table II we compare the objective value from our allocation to the one corresponding to the continuous solution of Lemma 4.7 (we report the percentage of traffic offloaded). As the latter is an upper bound on the optimal solution of the mixed-integer problem, the actual performance gap is bounded by the values shown in Table II. We refer to this policy as QoE-Aware Caching (QAC).

TABLE II: Estimated offloading gains of rounded allocation vs. continuous relaxation for different cache sizes (in percentage of the catalogue size).

Cache size	0, 1%	0, 2%	0, 5%	1%
Rounded (QAC)	34,25%	44,10%	52,88%	60,75%
Continuous	34,29%	44,12%	52,89%	60,75%



(a) Number of replicas $\mathbf{x}$. (b) Deadlines $\mathbf{y}$ (in s).

Fig. 3: Example of an allocation for *QAC* (semilog scale).

In Fig. 3, we show an example of the allocation provided by the *QAC* policy. In this example, the content catalogue is of 1000 content items with power-law popularity and random content size, the number of vehicles h is 500 and the cache capacity per vehicle is equal to the 0,5% of the catalogue. As expected, due to the skewed content popularity, the policy assigns *a lot* of copies to *a few* contents (Fig. 3a). However, the content size introduces some randomness in the number of replicas, i.e., contents with more replicas are not necessarily the most popular ones.

V. Specific policies for content of small size

In this section, we discuss a model that provides a tighter approximation of the problem when a content (of small size) can be *entirely* downloaded from the vehicular cloud during a *single* contact. This scenario can be considered reasonable when: (i) content size is small (e.g., short videos, news, ads); (ii) contact duration is large due to future envisioned improvements in Vehicle-To-Device communications. We formulate the corresponding offloading optimization problem in Section V-A and we propose two specific policies to optimally cache small popular content in the vehicular cloud when deadlines are variable (Section V-B) or fixed (Section V-C).

A. Optimization problem for content of small size

The optimization problem can be reformulated as follows:

Problem 2. *Let a content be entirely downloaded during a contact with high probability. The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud:*

$$\underset{\mathbf{x} \in X^k, \mathbf{y} \in Y^k}{\textit{maximize}} \ \Phi_{qac-sc}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - e^{-\lambda \cdot x_i \cdot y_i}), \quad (7)$$

subject to the same constraints as Problem (1)*.*

Proof. Given the previous assumption, the number of bytes downloaded from the cellular infrastructure for content i is equivalent to the probability to get the content i within y_i

multiplied by its size. We have seen in the proof of Proposition 4.2 that the superposition of the processes formed by the inter-meeting times with vehicles storing a content i (denoted by $\{T_i(t), t > 0\}$) approaches a Poisson process. Thus, the following equivalence holds:

$$\Phi_{qac-sc} = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - \mathbf{P}[T_i > y_i]) = \sum_{i=1}^{k} \phi \cdot s_i \cdot \left(1 - e^{-\lambda \cdot x_i \cdot y_i}\right).$$

We can replace the expression of Φ_{qac-sc} in Eq. (1) to obtain the objective function of the lemma. □

B. QoE-Aware Caching for Small Content (QAC-SC)

Problem (2) is a MINLP. Similarly to Section IV-B, we consider a continuous relaxation of the problem.

Corollary 5.1. *Problem* (2) *is a biconvex optimization problem with separable constraints.*

Proof. Eq. (7) is a twice-differentiable function on the variables $\{\mathbf{x}, \mathbf{y}\}$. In order to analyze the convexity of the function, we need to examine its matrix of second partial derivatives $H(\mathbf{x}, \mathbf{y})$. We find that $\det|H| = (2 \cdot \lambda \cdot x_i \cdot y_i - 1) \cdot e^{-\lambda \cdot x_i \cdot y_i}$ which is greater than 0 when $x_i \cdot y_i > \frac{1}{2\lambda}$. Since we can found pairs which makes the determinant of the Hessian negative, we have proved that the function is *not* convex. Rather, we note that Φ_{qac-sc} is convex on X^k for each $\mathbf{y} \in Y^k$ and convex on Y^k for each $\mathbf{x} \in X^k$. Thus, the objective function is *biconvex*. Since the constraints are linear and the feasible regions for the control variables are convex, the problem is biconvex. □

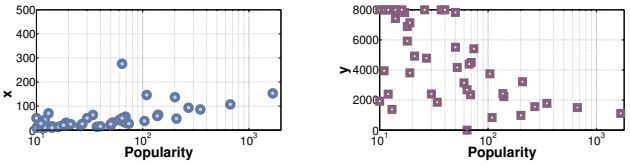
Different from convex optimization, a biconvex problem is a non-convex problem which may have a large number of local minimum points, and thus not easy to solve. Theoretically, its convex substructure can be exploited to solve such a problem as proposed by Floudas *et al.* [30]. However, their *global optimization* algorithm does not scale to our scenario since it requires to solve 2^k nonlinear subproblems in each iteration to obtain a new lower bound to the problem. As an alternative, we propose the *Multi-Start Alternate Convex Search* algorithm (Algorithm 1) that modifies the one described by Wendell and Hurter [31]. In our algorithm, at every step, only the variables of an active block are optimized while those of the other block are fixed. Since the resulting subproblems are convex, convex minimization methods can be used to solve them efficiently: we use *Lagrangian relaxation* which is well suited to limited resource allocation problems. Here the details of the algorithm:

1) Let $\mathbf{y}^0 \in Y^k$ denote an arbitrary initial feasible set of solutions for Problem (2).
2) Solve the following convex nonlinear problem:

$$\mathbf{x}^0 \leftarrow \max_{\mathbf{x} \in X^k} \sum_{i=1}^{k} \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i \cdot y_i^0}, \tag{8}$$

subject to the same constraints as in Problem (1). The solution can be easily found through the Lagrangian multiplier method, and gives

$$x_i^0 = \frac{1}{\lambda \cdot y_i^0} \cdot \ln\left(\frac{\lambda \cdot y_i^0 \cdot \phi_i}{\rho_x}\right), \tag{9}$$



(a) Number of replicas $\mathbf{x}$. (b) Deadlines $\mathbf{y}$ (in s).

Fig. 4: Example of an allocation for *QAC-SC* (semilog scale).

where ρ_x is an appropriate Lagrangian multiplier.
3) The set $\mathbf{x}^0$ is used as input for the same convex nonlinear problem optimized for $\mathbf{y} \in Y^k$. Similarly,

$$y_i^1 = \frac{1}{\lambda \cdot x_i^0} \cdot \ln\left(\frac{\lambda \cdot x_i^0 \cdot s_i^2}{\rho_y}\right), \tag{10}$$

where ρ_y is an appropriate Lagrangian multiplier.
4) If the stopping criterion (see below) is satisfied, then stop and $\{\mathbf{x}^0, \mathbf{y}^1\}$ is the solution. Otherwise, go to 2) such that $\mathbf{y}^0 \leftarrow \mathbf{y}^1$.

There are several ways to define the stopping criterion in Step 4). For example, one can consider the absolute value of the difference of the objective function comparing the vectors of solutions $\{\mathbf{x}^0, \mathbf{y}^0\}$ and $\{\mathbf{x}^0, \mathbf{y}^1\}$. Moreover, the order of Step 2) and Step 3) can be permuted. Since every iteration of the algorithm produces a partial optimum solution [32], we iterate the procedure described above for different arbitrary initial feasible sets, and we select the vectors $\{\mathbf{x}, \mathbf{y}\}$ that maximize Eq. (7). The accuracy of the solution depends on the number of iterations and on the parameter ϵ chosen to stop the search. While there is still no *theoretical* guarantee about the convergence to the optimal solution, this version of the algorithm can reach the global optimum with large probability. Additionally, it is possible to use a cutting-plane algorithm to eventually generate global optimal solutions. We refer to this policy as QoE-Aware Caching for Small Content (QAC-SC).

Algorithm 1 Multi-Start Convex Search Algorithm

```
Ensure: x, y
 1: function MAIN
 2:     max_f ← 0
 3:     output ← {∅, ∅}
 4:     for i ← 1 to max_iter do
 5:         y0 ← arbitrary feasible solution
 6:         x0 ← Eq. (9)
 7:         y1 ← Eq. (10)
 8:         while f(x0, y1) - f(x0, y0) > ϵ do
 9:             y0 ← y1
10:             x0 ← Eq. (9)
11:             y1 ← Eq. (10)
12:         if f(x0, y1) > max_f then
13:             output ← x0, y1
14:     return output
```

In Fig. 4, we show an example of the allocation provided by the *QAC-SC* policy with the same setup used in Fig. 3. Differently from *QAC*, this policy tends to assign a lower number of copies to a larger number of contents. This reflects the fact that the policy assumes a content to be downloaded during a single contact: once the probability to meet a vehicle is large enough (depending to the number of copies in the cloud), *QAC-SC* privileges less popular content which have not been cached yet.

C. Content offloading with fixed deadlines (FIXED)

Here we assume that deadlines are fixed for all content. This scenario is interesting for three main reasons: (i) fixed deadlines are very common in literature [22], and our model also includes vehicles mobility; (ii) it is possible to obtain closed-form results which are easy to analyze from an analytical point of view; (iii) this scenario will be used as baseline in the simulation section in order to evaluate the improvements provided by variable deadlines.

In this policy, we set $y_i = y^0 \ \forall i \in K$ (fixed deadlines) such that the QoE constraint of Eq. (3) is satisfied. Hence, the objective function of the optimization problem can be rewritten as in Eq. (8). The related optimization problem is a knapsack bounded problem with a nonlinear objective function, and is thus NP-hard. We solve the continuous relaxation of the problem to obtain a closed-form real-valued solution, which then we discretize through probabilistic rounding.

Theorem 5.2. *Let $y_i = y^0 \ \forall i \in K$ denote equal deadlines such that Eq. (3) is satisfied. Then, the optimal number of replicas that solves Problem* (2) *is given by*

$$x_i^* = \begin{cases} 0, & \text{if } \phi_i < L \\ \frac{1}{\lambda \cdot y^0} \cdot \ln\left(\frac{\lambda \cdot y^0 \cdot \phi_i}{\rho}\right), & \text{if } L \leq \phi_i \leq U \\ h, & \text{if } \phi_i > U \end{cases}$$

where $L \triangleq \frac{1}{\lambda \cdot y^0}$, $U \triangleq \frac{e^{h \cdot \lambda \cdot y^0}}{\lambda \cdot y^0}$, and ρ is an appropriate Lagrangian multiplier.

Proof. The proof can be found in Section IX. □

We refer to this policy as FIXED.

VI. Performance evaluation

We build a simulator based on real traces to evaluate the performance of our proposed policies and to compare it with related work. As for the vehicle mobility, differently than other traces either related to freeways and peripheral routes or focusing on microscopic mobility, we consider the mobility of taxis within San Francisco city center. Hence, our customized tool is specifically designed for a downtown type of urban envirnoments, as targeted by the paper. The implementation details are given in Section VI-A. Then, we compare different allocation strategies concerning the amount of data offloaded (Section VI-B), the user QoE (Section VI-C) and other architectures (Section VI-D).

A. Simulator description

The tool simulates YouTube video requests in the centre of San Francisco over a period of a few days. We develop a modular simulator with four main building blocks (Fig. 5): input data, caching policies, core algorithm, output. In the rest of the subsection we discuss in detail each of these blocks and how they are linked each other.

1) Input data: The inputs of the simulator are network parameters and (real or synthetic) traces for mobility and content popularity. In our simulations, we show the results based on real traces (except for user mobility), although we observed similar outcomes for synthetic ones. Specifically:

- *Vehicle mobility.* We use the Cabspotting trace [33] to simulate the vehicle behaviour; this trace records the GPS coordinates for 531 taxis in San Francisco with granularity of 1 minute. To improve the accuracy of our simulations, we increase the granularity to 10 seconds by linear interpolation. We also use this trace to extract the necessary mobility statistics for our model (e.g., the mean inter-meeting rate).
- *User mobility.* We use synthetic traces based on SLAW mobility model [34]. Specifically, according to this model, users move in a limited and defined area around popular places. The mobility is nomadic where users alternate between pauses (heavy-tailed distributed) and travelling periods at constant (but random) speed.
- *Content.* We infer the number of requests per day from a database with statistics for 100.000 YouTube videos [35]. The database includes static (e.g., title, duration) and dynamic information (e.g., views, shares, comments). In order to increase the number of simulations and to provide sensitivity analysis for content size, buffer capacity and cache density, we randomly select 10.000 contents from the catalogue. Content size is generated from either a truncated normal or a bounded Pareto distribution[5] (instead of using the content size from the YouTube trace) in order to experiment different characteristics of the catalogue.

We set $r = 1$ Mbps which approximates the playback of a 720p video (remember that r corresponds to the playout rate in the case of videos - see Assumption A.5). We set the cache size per vehicle in the range $0,1 - 1\%$ of the total catalogue which is an assumption that has also been used in other works [5], [23] (we use $0,2\%$ as a default value). Finally, we consider $\omega_{max} = 3$ which corresponds to an average deadline of *only* a few minutes (compared to video durations up to 1,5 hours).

2) Caching policies: The role of the *caching policies* block is to compute the number of replicas and the deadline per content given popularity and network parameters. The following allocation policies will be compared in the rest of the section:

- *QAC.* This policy solves the optimization problem with a reasonable approximation for content of generic size. This policy is described in Section IV-B.
- *QAC-SC.* This policy solves the optimization problem when a content can be downloaded with large probability in one contact. This policy is suitable for content of small size, and is described in Section V-B.
- *FIXED.* This policy solves the optimization problem when a content can be downloaded with large probability in on contact, and deadlines are fixed. This policy is described in Section V-C.
- *MP.* This policy stores the most popular content in vehicle buffers until caches are full while any other content gets 0 copies. Deadlines are fixed. This policy is optimal for sparse scenarios where caches do not overlap.

[5]Since content size and popularity are not correlated (from the analysis of the trace), we randomly assign content size to the catalogue.

TABLE III: Parameters used in the simulations.

Param	Value	Param	Value
h	531 vehicles	c	$0,2\% \cdot k$
k	10.000 contents	$\mathbf{E}[s]$	50-200 MB
r	1 Mbps (720p)	ω_{max}	3
y^0	~9 minutes	y_{max}	$10 \cdot y^0$
λ_{sr}	0,964 day^{-1}	λ_{lr}	2,83 day^{-1}

- *RAND.* Content is allocated randomly. Deadlines are fixed.

3) Core algorithm: The simulator implements a time slotted system where each iteration is equivalent to 10 seconds. During each time slot, a set of users request some contents. In practice, the tool links each content request to a user (characterized by a path according to the SLAW model aforementioned). This link is randomly generated as we assume that the correlation between locations and content requests is low which is a realistic assumption in small areas. We also build a *request generator* to assign a timestamp to each request[6]: inter-arrival times between successive requests are exponentially distributed according to the IRM model [36] which is the de facto standard in the analysis of storage systems.

The *core algorithm* is made up of two modules:

- *Content download module.* Since better signal strength implies higher signal-to-noise ratio at the receiver (and thus less errors), wireless data rate is indeed proportional to the signal strength [37]. Although other factors affect the signal strength (e.g., interference, physical barriers), distance plays a main role. For this reason, as most wireless protocols implement some *rate adaptation* mechanism, our simulator also varies the communication rate according to the distance between the user and the vehicle she is downloading from, with a *mean* of 5 Mbps. Inline with proposed protocols for vehicle communications (e.g., 802.11p, LTE ProSe), we set the maximum communication range between $\mathcal{U}$ and $\mathcal{H}$ nodes to 100 m (short range) or 200 m (long range).
- *Mobility module.* The goal of this module is to determine the vehicles to which a user can connect at a given time and the potential download rate. For each iteration, a function analyzes the user and vehicle mobility traces according to the parameters set by the content download module.

Given the setup described, a user is allowed to download a certain number of bytes while being in the communication range of a vehicle storing the requested content (this information is provided by content download + mobility modules). When the deadline expires, the potential remaining bytes are assumed to be downloaded from the cellular infrastructure.

4) Output: We generate content requests over a period of 5 days. For each request, we take note of the amount of bytes offloaded and the time needed to completely download the content. We define the following metric used to compare the different caching strategies:

Definition 6.1. *We refer to offloading gain as the sum of bytes offloaded over the total number of bytes requested.*

We summarized the main parameters in Table III.

[6]We assume that content requests are concentrated at day-time.

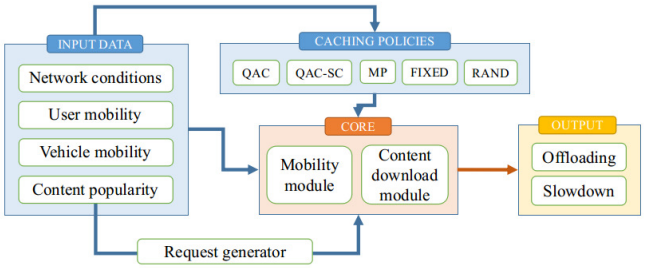


Fig. 5: Simulator building blocks.

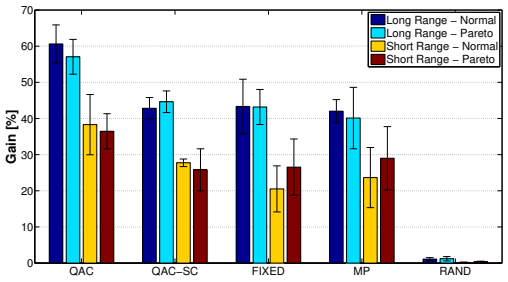


Fig. 6: Offloading gains when $\mathbf{E}[s]$ = 200MB.

B. Caching policies evaluation

In Figs. 6-7 we plot the offloading gains for different allocation policies given the parameters listed in Table III. Mean content size is of 200 MB (Fig. 6) and of 50 MB (Fig. 7). These plots also include the 95% confidence interval. From the analysis of the plots, we get the following findings:

- For large content, *QAC* offloads a much larger fraction traffic than any other policy in any situation (additional gains of around 20%). For instance, when long range communications are considered, offloading gains are in the order of 60% for *QAC*, and no more than 40% for *QAC-SC*, *FIXED* and *MP*. This is due to (i) variable deadlines and (ii) a model dealing with partial downloads.
- Although *QAC-SC* is expected to benefit from the deadline variability, this policy performs similar to fixed deadline policies since the assumption that a content can be downloaded in one contact is unrealistic for a content of 200 MB.
- As mean content size decreases, *QAC-SC* outperforms the other policies since its model is tailored to small content.
- *QAC* still performs better than fixed deadline policies, confirming the fact that its model provides a reasonable approximation for any content size.
- Not substantial differences have been observed for different

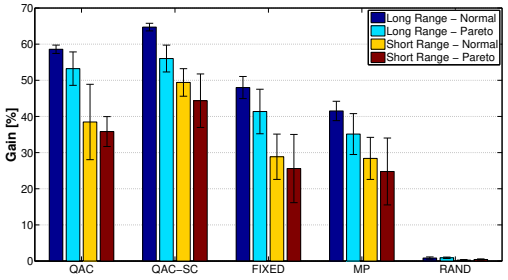


Fig. 7: Offloading gains when $\mathbf{E}[s]$ = 50MB.

content size distributions: however, from additional experiments we have noticed that, as the coefficient of variation of the content size distribution decreases (i.e., contents have similar size), the percentage of traffic offloaded by variable and fixed deadline policies becomes similar.
- A random policy perform poorly in any scenario due to the skewness of the content popularity.

Fig. 8 depicts the fraction of data offloaded by the vehicular cloud as a function of number of vehicles, buffer size and mean content size for long range communications when content size distribution is truncated normal. Specifically, in Fig. 8a we perform sensitivity analysis according to the number of vehicles h in the cloud which varies from 100 to 500. When h is larger than 200, more than 40% of the traffic can be offloaded by *QAC*. While the number of envisioned connected vehicles in the centre of San Francisco is expected to be much larger, the low technology penetration rate analyzed still provides considerable amount of data offloaded. This result is important to promote the start up phase of the vehicular cloud. However, it is interesting to note that in a sparse scenario ($h = 100$), *QAC* performs poorly. This happens because the value of $\mathbf{E}[B_i] = \lambda \cdot \mu \cdot x_i \cdot y_i$ that has been used in *QAC* holds only if the number of vehicles participating in the vehicular cloud is large (see Theorem 4.2). What is more, from Corollary 4.4, the error of the approximation used by *QAC* is proportional to the standard deviation of B_i which increases in a sparse environment.

Fig. 8b compares different buffer capacities per vehicle. Buffer size goes from 0,1% to 1% of the catalogue (where $h = 531$). Interestingly, considerable performance gains can be achieved with very reasonable storage capacities. Here the simulations are performed on a set of 10.000 contents, but in a scenario with a larger realistic catalogue (e.g., 1000 times larger), it seems doable to store 0,1-0,5% of the contents needed to achieve good savings. E.g., if one considers the entire Netflix catalogue (~3PB), a mobile helper capacity of about 3 TB (0,1%) already suffices to offload more than 40% of the total traffic for long range communications (while around 30% for fixed deadline policies). What is more, our simulations consider a very low technology penetration rate where only 1% of the expected number of vehicles in San Francisco is part of the vehicular cloud. However, when the number of caches grows, buffer capacities can be smaller to achieve (at least) the same offloading gains. As a final note, as the cache capacity increases, *QAC-SC* offloads much more traffic than *FIXED*, while this is less evident when the cache size per vehicle is lower. Basically, as the cache size increases, offloading gains are mainly provided by the deadline variability rather than the cache policy chosen.

In Fig. 8c we analyze the effect of content size by varying the mean content size from 30 MB to 200 MB. As expected, for small content (say for $\mathbf{E}[s] < 80$ MB), *QAC-SC* offloads more traffic than any other policy. After this threshold, since the assumption of entire download of a content during a contact becomes inaccurate, this policy offloads less traffic. A similar behavior can be seen for *FIXED* that exploits the same assumption. What is important to notice, however, is that the traffic offloaded by *QAC* is quite stable for any content size.

C. QoE analysis

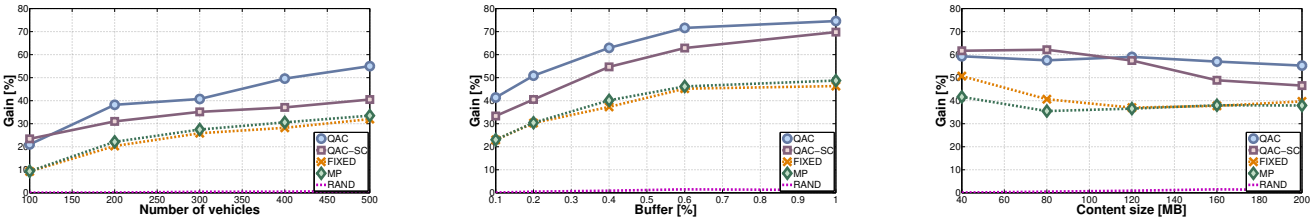
In this subsection, we perform an analysis of the user QoE by allowing different values of ω_{max}. In Fig. 9, we show the upper bound on the mean slowdown ω_{max} that an MNO should set in order to reach some specific offloading gains, from 30% to 60%. We consider long range communications, and content size drawn from a truncated normal distribution with mean 200MB, but similar results can be obtained for short range communications or other content size distributions. The required mean slowdown to offload more traffic increases slowly for variable deadline policies while we notice an exponential growth for fixed deadlines. Basically, Fig. 9 can be seen as a description of the effect produced by additional gains on the QoE: for instance, an MNO should double the value of ω_{max} (100% increase) with *FIXED* policy to offload 10% more traffic, while the mean slowdown only increases in the range of 15-40% for *QAC* and *QAC-SC* to have the same improvement in the offloading gains. This low impact on the slowdown highlights the advantages introduced by our QoE-aware policies. Knowing the function that ties user experience and slowdown (e.g., linear, logarithmic) can lead to a better interpretation of the plot. However, this behavioural analysis goes beyond the scope of the paper.

D. Mobile vs. static helpers

We refer to the well-known femtocaching framework described in Golrezaei *et al.* [4] as the state-of-the-art in mobile data offloading. The goal of the authors is to optimally store content in a distributed cloud storage built with SCs to maximize the traffic offloaded. Instead, our proposed caching policies are modeled according to a futuristic cloud based on mobile helpers, which brings the traditional femtocaching framework a step further. Our claim is that vehicle speed can be successfully exploited to increase the percentage of traffic offloaded by the caches. In this subsection, we perform simulations based on the Cabspotting trace to validate the above conjecture.

In our simulations, we build the femtocaching environment such that *the estimated CAPEX and OPEX are comparable to the costs needed for our vehicular cloud*. A cost analysis based on [38] (which we omit due to space limitations, and can be found in [39]) estimates that our architecture can introduce a ten-fold cost reduction compared to small cells. Therefore, we consider a sparser deployment with only 53 SCs (note that we have 531 vehicles) such that the total cost is equalized. SC helpers are distributed in the considered area proportionally to the popularity density, i.e., areas with a higher number of requests have higher SC density (this is a common operator policy since SCs are deployed to alleviate traffic "hotspots"). As previously described, users move according to the previously described SLAW trace, and they can also download video chunks at low cost from a nearby SC if it stores the requested video.

Fig. 10 compares the offloaded gains for vehicular cloud and the femtocaching implementation described above. As expected, gains provided by the vehicular cloud are considerably higher than femtocaching for both short and (mainly)



(a) Offloading gain vs. h ($c = 0{,}2\% \cdot k$). (b) Offloading gain vs. c ($h = 531$). (c) Offloading gain vs. mean content size.

Fig. 8: Fraction of traffic offloaded as a function of vehicle density (Fig. 8a), buffer capacity (Fig. 8b) and mean content size (Fig. 8c) for long range communications.

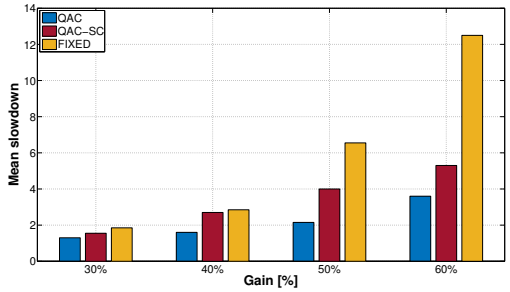


Fig. 9: Mean slowdown due to reach specific offloading gains.

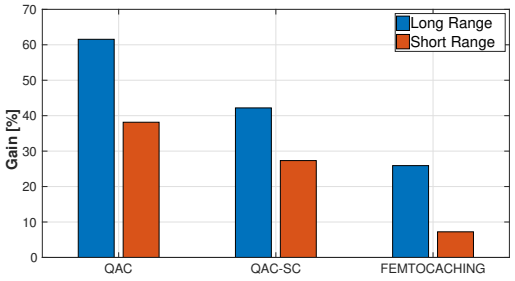


Fig. 10: Gains for vehicular cloud and femtocaching.

long range communications. This plot reveals the potential of an additional low cost infrastructure made up of vehicles. This study can be considered relevant since it provides some evidence of the potentiality of the vehicular cloud, and may speed up its adoption.

VII. Discussion

A. Communication protocol

The recent increasing interest in vehicular networks has led to the proliferation of new standards and protocols for high mobility environments. The IEEE 802.11p protocol, which has been developed for the specific context of vehicular networks, is considered as the de facto standard. It includes physical and MAC layer specification as well as upper-layer protocols. Specifically, IEEE 802.11p is expected to be particularly suitable for medium range communications and delay-sensitive applications. According to the modulation used, throughput can be from 2-3 Mbps (with BPSK up to 150 m) to 15-20 Mbps (with 64QAM up to 25 m). While this protocol actually covers simplicity (uncoordinated access mechanism, no authentication) and low delay (few hundreds milliseconds in crowded areas), its decentralized nature imposes limitations on reliability, congestion (due to higher beaconing frequency), and scalability. Concerning battery drain, it has also been shown that it is possible to implement a low battery consumption version in modern mobile devices without compromising performance. Given the diverse performance requirements from a wide spectrum of vehicular networking applications, recently several standardization bodies and research consortium have shown increasing interest in adopting LTE Advanced to support device-to-device communications and vehicular network applications. Specifically, 3GPP Release 12 has introduced Proximity Services (ProSe) for LTE Advanced which envisages two basic functionalities: *ProSe discovery* that identifies the ProSe-enabled devices in proximity and *ProSe communication* that enables establishment of communication paths via PC5 interface between two or more ProSe-enabled devices that are in direct communication range.

B. Business model and incentives

While the number of cars with some sort of networking ability today is small, it is estimated that around 90 percent of all manufacturers' new models are likely to have Internet connectivity by 2020. For instance, BMW, that has already been embedding SIM cards for mobile connectivity in all its new cars, has recently unveiled the Vehicular CrowdCell project where a mobile femtocell optimizes the mobile radio reception inside vehicles and is also capable to enhance the capacity and coverage of mobile radio networks. Specifically, cellular operators see the connected car as another device to be hooked up to their networks, and they have started to propose data plan dedicated to vehicles (e.g., AT&T in United States). Cellular operators might offer economic incentives (e.g., subscription reduction) to users that decide to join the vehicular cloud with their private vehicles. This should lead to a double benefit, thus increasing their market share by offloading part of the mobile traffic. What is more, modern cities might decide to install these cheap devices into buses or trams to provide additional services. An interesting example is given by Portugal where the company Veniam has recently built the largest vehicular network in the world [9]. Specifically, they can offer Wi-Fi features in public transportation, increasing number of passengers, reducing emissions and generating additional revenue. Furthermore, vehicular networks can produce real-time city-scale data from cheap sensors which can be used to increase safety and efficiency of municipal operations.

C. Cache refresh

MNOs periodically update their caches (e.g., every two days, once a week) when cellular infrastructure is underloaded (i.e., at night time or off-peak hours) with incremental changes, keeping the vehicular cloud up-to-date. However, caching is optimized only if a fresh view of the system is maintained, but content popularity prediction can be a challenging task because of its time-varying nature. In our work, we have assumed content popularity to be stable in the time interval considered. While this is not true in general, some contents (e.g., software updates, YouTube videos) show a quite stable behaviour, making this assumption a good approximation. What is more, prediction techniques for video popularity based on history are accurate for short-medium terms.

We propose two more options to manage varying popularity:

- *Randomized seeding time window.* Content seeding is temporally shifted for different vehicles, or distributed solutions where the decision to cache or not is left to each vehicle.
- *Dynamic adaption of content popularity.* Caches are updated dynamically as new contents are introduced in the catalogue, and/or existing contents exhibit a significant change in popularity. Adapting to changing content popularity is not only important to introduce new contents and delete obsolete ones, but also to increase the potential performance gains. The details are described in our previous work [22].

VIII. Conclusion and future work

Compared to similar works in mobile edge computing, this work introduces several contributions: (i) it considers mobile relays (vehicles) that virtually increase the cache size seen by pedestrian users; (ii) while the majority of the works consider fixed deadlines, our paper deals with variable TTLs by introducing a QoE metric; (iii) the generic model includes partial downloads from vehicles. In this paper, we propose caching policies that can be exploited by MNOs in different contexts and scenarios. These policies have been largely validated analytically and through real trace simulations. The comparison with traditional approaches shows a large increment in the percentage of traffic offloaded. We have also given insights to an operator on how to correctly choose the policy to use, and how to set the QoE parameters.

As future work, it would be interesting to tune the user QoE taking into account the content *type* along with the content size. While we have shown that QAC performs well in the majority of the situations, it would be interesting to study closer approximations for the generic formulation of the problem. Also mixed policies to tie QAC to QAC-SC can probably bring additional gains. Furthermore, potential extensions of our current simulation framework may be helpful: for instance, one can use a realistic vehicular simulator to also take into account the potential overhead in the network layer and an improved interference model. Further developments may lead to the realization of a testbed implementation in real vehicles.

Acknowledgment

This work was funded by the French Government, through the EUR DS4H Investments in the Future project managed by the National Reseach Agency (ANR) with the reference number #ANR-17-EURE-0004.

IX. Additional proofs

A. Proof of Lemma 4.6

1) The following equalities hold:

$$\lim_{s_i \to 0} \Phi_{qac} = \lim_{s_i \to 0} \Phi = \sum_{i=1}^{k} \phi_i \cdot s_i$$

$$\lim_{s_i \to +\infty} \Phi_{qac} = \lim_{s_i \to +\infty} \Phi = \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}[B_i].$$

2) It is easy to see that

$$\mathbf{E}[\min\{B_i, s_i\}] = F_{B_i}(s_i)\mathbf{E}[B_i | B_i \leq s_i] + s_i(1 - F_{B_i}(s_i)). \quad (11)$$

$\mathbf{E}[B_i | B_i \leq s_i]$ corresponds to the truncated mean of B_i upper bounded by s_i. If the number of meetings within y_i is large, B_i can be considered as a normal distribution from Corollary 4.4. Thus, we can write its truncated mean as

$$\mathbf{E}[B_i | B_i \leq s_i] = \mathbf{E}[B_i] - \sigma_{B_i} \cdot \frac{f_{B_i}(s_i)}{F_{B_i}(s_i)},$$

where σ_{B_i} is the standard deviation of B_i, and can be inferred from Corollary 4.3[7]. If $\mathbf{E}[B_i] > s_i$, the error e introduced by $\Phi_{qac}(\cdot)$ can be evaluated as follows:

$$\begin{aligned} e &= \sum_{i=1}^{k} \phi_i \cdot |\min\{\mathbf{E}[B_i], s_i\} - \mathbf{E}[\min\{B_i, s_i\}]| \\ &= \sum_{i=1}^{k} \phi_i \cdot |s_i - \mathbf{E}[\min\{B_i, s_i\}]|. \end{aligned} \quad (12)$$

Then, we compute the second term of Eq. (12) from Eq. (11), and, after some calculations, we obtain

$$e = \sum_{i=1}^{k} \phi_i \cdot [F_{B_i}(s_i) \cdot |\mathbf{E}[B_i] - s_i| + \sigma_{B_i} \cdot f_{B_i}(s_i)].$$

We repeat the same steps for $\mathbf{E}[B_i] \leq s_i$.

B. Proof of Theorem 5.2

Proof. Note that the constraint on deadlines is satisfied by assumption. The value of y_0 can be directly inferred by Eq. (3) where, solving for $y_i = y^0$, we obtain:

$$y^0 \leq \frac{(\omega_{max} - 1)}{\sum_{i=1}^{k} \phi_i \cdot r / s_i} \cdot \sum_{i=1}^{k} \phi_i.$$

The derivative of the Lagrangian function of the problem is:

$$\frac{\partial \mathcal{L}}{\partial x_i^*} = -\lambda \cdot y^0 \cdot \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i^* \cdot y^0} + l_i - m_i - \rho \cdot s_i,$$

where l_i and m_i are appropriate Lagrangian multipliers related to the bounds of $\mathbf{x}$. According to the method of the Lagrangian

[7]Note that $\sigma_{B_i} \neq \sigma$, the latter being the standard deviation for a single contact.

multipliers, this equation must be equal to 0. Making explicit x_i^*, we obtain:

$$x_i^* = \frac{1}{\lambda \cdot y^0} \cdot \ln\left(\frac{\lambda \cdot y^0 \cdot s_i \cdot \phi_i}{s_i \cdot \rho - l_i + m_i}\right).$$

Finally, the system constraints create three regimes depending on the content popularity. □

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update," 2016-2021.

[2] Small Cell Forum, "Backhaul technologies for small cells: Use cases, requirements and solutions," 2013.

[3] X. Bao, Y. Lin, U. Lee, I. Rimac, and R. R. Choudhury, "Dataspotting: Exploiting naturally clustered mobile devices to offload cellular traffic," in *Proceedings IEEE INFOCOM*, pp. 420–424, April 2013.

[4] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless Device-to-Device Communications with Distributed Caching," *CoRR*, vol. abs/1205.7044, 2012.

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

[6] N. Alliance, "NGMN 5G White Paper." https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf, 2015.

[7] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, 2014.

[8] G. S. Paschos, S. Gitzenis, and L. Tassiulas, "The effect of caching in sustainability of large wireless networks," in *WiOpt*, pp. 355–360, 2012.

[9] "Veniam." https://veniam.com/.

[10] E. Lee, E. Lee, M. Gerla, and S. Y. Oh, "Vehicular cloud networking: architecture and design principles," *IEEE Comm. Magazine*, 2014.

[11] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting Mobile 3G Using WiFi," in *MobiSys*, pp. 209–222, ACM, 2010.

[12] F. Mehmeti and T. Spyropoulos, "Is it worth to be patient? Analysis and optimization of delayed mobile data offloading," in *IEEE INFOCOM Conference on Computer Communications*, pp. 2364–2372, April 2014.

[13] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile Data Offloading: How Much Can WiFi Deliver?," *IEEE/ACM Transactions on Networking*, vol. 21, pp. 536–550, April 2013.

[14] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. 1st ed., 2013.

[15] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, pp. 11339–11351, Dec 2017.

[16] P. Sermpezis and T. Spyropoulos, "Not All Content is Created Equal: Effect of Popularity and Availability for Content-centric Opportunistic Networking," in *MobiHoc*, pp. 103–112, ACM, 2014.

[17] G. Gao, M. Xiao, J. Wu, K. Han, and L. Huang, "Deadline-Sensitive Mobile Data Offloading via Opportunistic Communications," in *IEEE SECON*, pp. 1–9, June 2016.

[18] X. Li, X. Wang, P. Wan, Z. Han, and V. C. M. Leung, "Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design," *IEEE Journal on Sel. Areas in Comm.*, 2018.

[19] B. Han, P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile Data Offloading through Opportunistic Communications and Social Participation," *IEEE Trans. on Mobile Computing*, 2012.

[20] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: A Popularity Aware Content Sharing Scheme in VANETs," in *29th IEEE International Conference on Distributed Computing Systems*, pp. 223–230, June 2009.

[21] J. Zhao and G. Cao, "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks," *IEEE Trans. on Vehicular Technology*, 2008.

[22] L. Vigneri, T. Spyropoulos, and C. Barakat, "Storage on wheels: Offloading popular contents through a vehicular cloud," in *IEEE WoWMoM*, 2016.

[23] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Comm. Magazine*, 2013.

[24] G. Szabo and B. A. Huberman, "Predicting the Popularity of Online Content," *Communications of the ACM*, vol. 53, pp. 80–88, Aug. 2010.

[25] S. Karlin and H. Taylor, *A First Course in Stochastic Processes*. Elsevier Science, 2012.

[26] H. Schmidli, "Lecture Notes on Risk Theory."

[27] R. Kannan and C. L. Monma, "On the computational complexity of integer programming problems," in *Optimization and Operations Research*, pp. 161–172, Springer, 1978.

[28] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[29] P. Raghavan and C. D. Tompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[30] C. Floudas and V. Visweswaran, "A global optimization algorithm (GOP) for certain classes of nonconvex NLPs," *Computers & Chemical Engineering*, vol. 14, no. 12, pp. 1397 – 1417, 1990.

[31] R. E. Wendell and A. P. Hurter, "Minimization of a non-separable objective function subject to disjoint constraints," *Operations Research*, vol. 24, pp. 643–657, Aug. 1976.

[32] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: a survey and extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[33] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "DAD data set epfl/mobility (v. 2009-02-24)." http://crawdad.org/epfl/mobility/, Feb 2009.

[34] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "SLAW: A New Mobility Model for Human Walks," in *IEEE INFOCOM*, 2009.

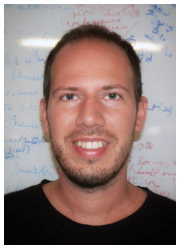
[35] "YouStatAnalyzer database." http://www.congas-project.eu/youstatanalyzer-database.

[36] E. G. Coffman and P. J. Denning, *Operating systems theory*, vol. 973. Prentice-Hall Englewood Cliffs, NJ, 1973.

[37] J. P. Pavon and S. Choi, "Link adaptation strategy for ieee 802.11 wlan via received signal strength measurement," in *IEEE ICC*, vol. 2, pp. 1108–1113 vol.2, May 2003.

[38] M. Paolini, "The economics of small cells and wi-fi offload," *Senza Fili Consulting, Tech. Rep*, 2012.

[39] L. Vigneri, *Vehicles as a mobile cloud : modelling, optimization and performance analysis*. Theses, Université Côte d'Azur, July 2017.



Luigi Vigneri obtained his Master in Computer Engineering from a joint program from Politecnico di Torino, Italy and Telecom ParisTech, France. He obtained his Ph.D in mobile communications at EURECOM, France. He was a post-doctoral researcher at Huawei Labs, France. He is currently senior research scientist at IOTA Foundation. His main research interests concern network modelling and optimization, and performance analysis of computer systems.

Thrasyvoulos Spyropoulos received the Diploma in Electrical and Computer Engineering from the University of Athens, and a Ph.D degree from the University of Southern California. He was a post-doctoral researcher at INRIA and then, a senior researcher at ETH, Zurich. He is currently an Assistant Professor at EURECOM, Sophia-Antipolis. He is the recipient of the best paper award in IEEE SECON 2008, and IEEE WoWMoM 2012, and runner-up for ACM Mobihoc 2011, and IEEE WoWMoM 2015.

Chadi Barakat is permanent researcher at INRIA Sophia Antipolis. He got his master, PhD and HDR degrees in Computer Science from the University of Sophia Antipolis. He was at EPFL-Lausanne for a post-doctoral position, and with Intel Research Cambridge as visiting faculty. He is currently on the editorial board of Computer Networks. His main research interests are in Internet measurements and traffic analysis, user QoE and network transparency, software-defined and mobile wireless networking.