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

Luigi Vigneri
EURECOM
450 Route des Chappes
Biot, France 06410
luigi.vigneri@eurecom.fr

Thrasyvoulos Spyropoulos
EURECOM
450 Route des Chappes
Biot, France 06410
spyropou@eurecom.fr

Chadi Barakat
INRIA Sophia Antipolis
2004 Route des Lucioles
Valbonne, France 06902
chadi.barakat@inria.fr

## 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 a variable deadline policy; (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.

## 1 INTRODUCTION

The large diffusion of handheld devices is leading to an exponential growth of the mobile traffic demand which is already overloading the core network [7]. To deal with such a problem, several works suggest to store content in small cells (SCs) or user equipments. Recently, it has been proposed the use of private or public transportation as storage points and mobile relays (*vehicular cloud*) [1, 2, 24] 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 cost involved make this an interesting alternative.

In this paper, we exploit such a vehicular cloud to store popular content to offload part of the mobile traffic demand. In our model, a user can query nearby vehicles to download a content with low delay (and at low cost for the MNO). However, since caches will be quite small compared to the daily catalogue, the user might not be inside the 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. Conversely, if the content is not found within a deadline, the user will be redirected to the cellular infrastructure. 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.* Most of related works [4, 6, 19] *require the user to move* 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, such delayed offloading architectures require deadlines in the order of half to a couple of hours to demonstrate performance benefits [4, 18, 19]. Instead, when caches are on vehicles, especially in a dense urban environment, a 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.

*Variable deadlines.* The majority of edge caching related works aims at policies that exclusively minimize the load on the cellular infrastructure. In most delayed offloading settings, the worst-case delay guarantee offered to the user is usually *fixed* for all content requests and set to large values. Conversely, in this work we allow the operator to *set different deadlines for different content.* This variability in the deadlines 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) to improve user Quality of Experience (QoE) as we explain below.

*User QoE-Aware offloading.* We evaluate the user QoE according to the experienced *slowdown* which has recently become popular [13]. This metric relates the waiting delay with the "net" download time. For example, a user requesting a web page of a few megabytes (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. 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, the main focus of this paper is on the modelling of the above scenario and on the formulation of a related (nontrivial) optimization problem. The main contributions of the paper can be summarized as follows:

- We model the problem of maximizing the percentage of traffic offloaded through the vehicular cloud considering the user QoE (captured by the slowdown metric) and a large range of realistic conditions (e.g., content of heterogeneous size), and we solve the corresponding optimization problem.

- We validate our findings using simulations with real traces for vehicle mobility and content popularity. We show that, in an urban scenario, our system can achieve considerable offloading gains with modest technology penetration (less than 1% of vehicles participating in the cloud) and low mean slowdown (that leads to average deadlines of a few minutes).
- We study the impact of different user QoE guarantees on operator- and user-related performance, and compare variable and fixed deadline policies.

The rest of the paper is organized as follows: in Section 2, we compare our work with the previous literature; in Section 3, we define the system model with the main assumptions; then, in Section 4, we present the mathematical formulation of the problem, and we solve a reasonable approximation (since the original problem is hard); we validate our results through real trace-based simulations in Section 5; finally, we conclude our paper with a summary and future work in Section 6.

## 2 RELATED WORK

Caching at the edge of the network has been deeply investigated by researchers lately [11, 21]. Golrezaei *et al.* [11] 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.* [21] 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. 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 by SCs, which comes at a high cost [3]. 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 the smaller number of users per cell raise the question whether enough overlap in user demand would be generated locally to have a high enough hit ratio, when real traffic is considered.

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 [4, 19], or until to reach peer nodes in a P2P infrastructure [6]. For example, Balasubramanian *et al.* [4] 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 [6, 10] 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, 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 delays

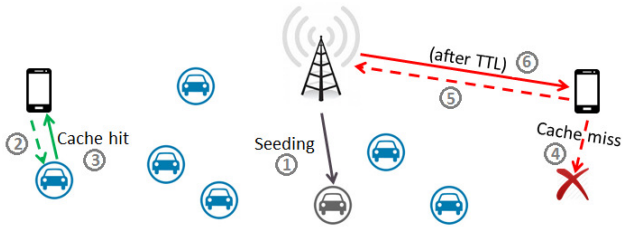



**Figure 1: Communication protocol.**

(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, 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 a previous work, we have dealt with the idea of vehicular cloud used to offload part of the traffic and accessible by handheld devices [24]. 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 car manufacturers [1] or by the launch of new companies [2] that offer network connectivity to public and private transportation.

## 3 SYSTEM MODEL

### 3.1 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):

- ($\mathcal{I} \rightarrow \mathcal{H}$): $\mathcal{I}$ nodes place content in $\mathcal{H}$ nodes according to the chosen allocation policy. This policy is the main outcome of this paper. We refer to this phase as *seeding* which 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, the seeding cost can be considered equal to 0. In our work we will focus on this scenario[1].
- ($\mathcal{H} \rightarrow \mathcal{U}$): an end user node can request content $i$ to the vehicles that are inside her communication range[2]. If content $i$ is found,

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

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 allocation policy during the seeding phase. The related local access cost is null.
- $(\mathcal{I} \rightarrow \mathcal{U})$: in case of a content not successfully downloaded within a time $y_i$, the $\mathcal{U}$ node's request will be served (partially or entirely) by the cellular infrastructure. The related cost is equal to the number of bytes downloaded from $\mathcal{I}$ nodes.

## 3.2 Main assumptions

**A.1 - Catalogue**. Let $\mathcal{K}$ be the set of all possible contents that users might request (also defined as "catalogue"), where $|\mathcal{K}| = k$. Let further $c$ be the size of the cache in each vehicle. We make the natural assumption that $c \ll k$. A content $i \in \mathcal{K}$ is of size $s_i$ (in MB) and is characterized by a popularity value $\phi_i$ measured as the expected number of requests within a seeding time window from all users and all cells.

**A.2 - Inter-meeting times**. We assume that the inter-meeting times $T_{ij}(t)$ 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 $\lambda$. 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 [15].

**A.3 - Cache model**. Let $x_i$ denote the number of vehicles storing content $i$. The vector $\mathbf{x}$ will be the control variable for our optimal cache allocation problem. We also assume $\mathcal{H}$ nodes to *store the whole content*, i.e., fractional storage is not allowed.

**A.4 - Chunk download**. 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 $\mu$ and variance $\sigma^2$. Let further $M_i$ be a random variable 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 $y_i$ for 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* imposed by our system when the content is fetched from the infrastructure that ties content download time to its size as $\omega_i \triangleq \frac{y_i + t_i}{t_i} = 1 + \frac{y_i}{s_i/r}$. The larger $\omega_i$ 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 $\omega_{max} > 1$ that upper bounds the mean slowdown. This value can be seen as a sort of "budget" available to the MNO that can be reallocated between contents. Moreover, it can set a maximum tolerable deadline $y_{max}$ to avoid excessively large deadlines for specific content.

We summarize the notation used in the paper in Table 1.

**Table 1: Notation used in the paper.**

| | |
|---|---|
| **CONTROL VARIABLES** | |
| $x_i$ | Number of replicas stored for content $i$ |
| $y_i$ | Deadline for content $i$ |
| **CONTENT** | |
| $k$ | Number of content in the catalogue |
| $\phi_i$ | 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 |
| $\lambda$ | 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 |
| $\mu$ | Mean of $b_{ij}$ |
| $\sigma^2$ | 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 infrastructure (or playout rate for videos) |
| $\Omega$ | Mean slowdown |
| $y_{max}$ | Maximum deadline |
| $\omega_{max}$ | Upper bound on the mean slowdown |

# 4 OPTIMAL CONTENT ALLOCATION

## 4.1 Offloading optimization problem

The operator's goal is to define a policy to maximize the bytes offloaded through the vehicular cloud while satisfying storage capacity and user QoE requirements. This policy should infer the optimal content allocation $\mathbf{x}$ and the optimal deadlines $\mathbf{y}$ to assign to the content catalogue. The number of bytes offloaded through the vehicular cloud *per request* 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. The following lemma captures these considerations in the objective function $\Phi(\mathbf{x}, \mathbf{y})$ to be optimized:

Lemma 4.1. *Given the previous assumptions, the amount of bytes offloaded through the vehicular cloud during a seeding time window*

*is given by*

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

Corollary 4.2. *The objective function* $\Phi(\mathbf{x}, \mathbf{y})$ *is equivalent to*

$$\Phi(\mathbf{x}, \mathbf{y}) \equiv \sum_{i=1}^{k} \phi_i \cdot \int_0^{s_i} \left(1 - F_{B_i}(t)\right) dt, \tag{2}$$

*where* $F_{B_i}$ *is the CDF of* $B_i$.

Proof. The objective function can be written as follows:

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

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. (2). □

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:*

$$\begin{aligned} \underset{\mathbf{x}\in X^k, \mathbf{y}\in Y^k}{\textit{maximize}} \quad & \sum_{i=1}^{k} \phi_i \cdot \int_0^{s_i} \left(1 - F_{B_i}(t)\right)\, dt \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \le c \cdot h, \\ & \Omega(\mathbf{y}) \le \omega_{max}, \end{aligned} \tag{3}$$

*where* $X \triangleq \{a \in \mathbb{R} \mid 0 \le a \le h\}$ *and* $Y \triangleq \{b \in \mathbb{R} \mid 0 \le b \le y_{max}\}$.

Each vehicle has a storage constraint and cannot store more than $c$ contents. However, instead of considering $h$ individual storage constraints, we only consider the global cache capacity of the vehicular cloud that corresponds to improve the tractability of the problem. Although the global capacity constraint introduces an error in the problem formulation, such an error is expected to be low when caches are large compared to the mean content size as we will explain at the end of this section (see randomized rounding).

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

Lemma 4.3. $B_i$ *can be approximated by a compound Poisson process as the number of vehicles participating in the vehicular cloud increases, if the mean inter-meeting rate with such vehicles is small.*

Proof. Assume that user $m$ requests content $i$. 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 $\lambda$ (see Assumption A.2). Let further $\{T_i(t), t > 0\}$ be the superposition of these processes. According to the Palm-Kintchine theorem [16], $\{T_i(t)\}$ approaches a Poisson process with rate $\lambda \cdot x_i$ if $x_i$ large[3] and $\lambda$ 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. □

Lemma 4.4. *The first two moments of* $B_i$ *are given by:*

$$\mathbf{E}[B_i] = \mu \cdot \lambda \cdot x_i \cdot y_i,$$

$$Var[B_i] = (\mu^2 + \sigma^2) \cdot \lambda \cdot x_i \cdot y_i.$$

Proof. The expected value of a compound Poisson process can be computed using conditional expectation, where the expectation is calculated using the Wald's equation. Similarly, it is possible to compute the moment of second order of $B_i$, and then its variance using the total law of variance. □

Lemma 4.5. *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. A *random* sum of identically distributed 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

$$B^*_i(s) = M^*_i(b^*_{ij}(s)),$$

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 Lemma 4.3), 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). □

[3] 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 [9].

All the quantities needed to solve the optimization problem are known, and can be plugged in Eq. (3). 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.

## 4.2 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.

Theorem 4.6. *Problem* (1) *is an NP-hard combinatorial problem.*

Proof. The problem is NP-hard since it includes mixed-integer linear programming as a subproblem [14]. □

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: first, it is possible to evaluate the quality of a feasible set of solutions; second, it is much faster to optimize than the mixed-integer problem. According to this relaxation, we also introduce a new objective function $\Phi_{qac}(\cdot)$ that approximates Eq. (1) in order to convert the problem in a convex optimization problem, hence improving tractability.

Lemma 4.7. *Eq. (1) can be approximated by*

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

Corollary 4.8. *Let* $e \triangleq \Phi_{qac} - \Phi$ *be the error introduced by Lemma 4.7. 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.,* $e$ *tends to* 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. (1) The following equivalences are true:

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

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

(2) It is easy to see that

$$\mathbf{E}[\min\{B_i, s_i\}] = F_{B_i}(s_i) \cdot \mathbf{E}[B_i | B_i \leq s_i] + s_i \cdot (1 - F_{B_i}(s_i)). \quad (5)$$

$\mathbf{E}[B_i | B_i \leq s_i]$ 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 [23]. 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 $\sigma_{B_i}$ is the standard deviation of $B_i$, and can be inferred from Corollary 4.4[4]. If $\mathbf{E}[B_i] > s_i$, the error $e$ introduced by $\Phi_{qac}$ can be evaluated as follows:

$$\begin{aligned} e &= \sum_{i=1}^{k} \phi_i \cdot |\min\{\mathbf{E}[B_i], s_i\} - \mathbf{E}[\min\{B_i, s_i\}]| \\ &= \sum_{i=1}^{k} \phi_i \cdot (s_i - \mathbf{E}[\min\{B_i, s_i\}]). \end{aligned} \quad (6)$$

Then, we compute the second term of Eq. (6) from Eq. (5), and, after some calculations, we obtain:

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

Similarly, we compute $e$ when $\mathbf{E}[B_i] \leq s_i$. □

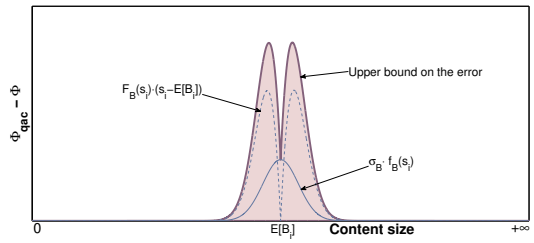



**Figure 2: Error introduced by $\Phi_{qac}(\mathbf{x}, \mathbf{y})$ in Lemma 4.7 for a fixed value of $\mathbf{E}[B_i]$.**

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* optimization problem that can be solved extremely efficiently and reliably:

Problem 2. *Consider the approximation introduced by Lemma 4.7. Then, the solution to the following convex optimization problem maximizes the bytes offloaded through the vehicular cloud:*

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

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

[4]Note that $\sigma_{B_i} \neq \sigma$ that is the standard deviation for a single contact.

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

$$\Phi_{qac}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{k} \phi_i \cdot \min\{\mathbf{E}[B_i], s_i\}$$
$$= \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}[B_i], \quad \text{s. t. } \mathbf{E}[B_i] \leq s_i, \ \forall i \in K, \qquad (7)$$

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 Lemma 4.4. According to Eq. (7), Problem (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[5] 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 [5]. 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 the objective function and of the constraints, it can be proved that the related problem is convex [5]. □

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* [22] 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 2 we compare the objective value from our allocation to the one corresponding to the continuous solution of Problem (2) (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 2. We refer to this policy as QoE-Aware Caching (*QAC*).

[5]A posynomial function $f(x)$ is a sum of monomials: $f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$, where $c_k > 0$.

**Table 2: 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% |

# 5 PERFORMANCE EVALUATION

## 5.1 Simulation setup

We build a trace-driven MATLAB simulator to validate our theoretical findings. Our tool simulates YouTube requests in the centre of San Francisco over five days. We use the following traces:

- *Vehicle mobility.* We use the Cabspotting trace [20] 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.
- *User mobility.* We use synthetic traces based on SLAW mobility model [17]. 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 [25]. 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.

Inline with proposed protocols for vehicle communications (e.g., 802.11p, LTE ProSe), we consider short (100 m) or long (200 m) communication ranges between $\mathcal{U}$ and $\mathcal{H}$ nodes. 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. We also set $r = 1$ Mbps which approximates the streaming 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 [12, 21] (we use 0,2% as a default value). We generate content size from either a truncated normal or a bounded Pareto distribution[6] (instead of using the content size from the YouTube trace) in order to experiment different characteristics of the catalogue. Finally, we consider $\omega_{max} = 3$ which corresponds to an average deadline of *only* a few minutes (compared to video durations that can go up to 1,5 hours).

Our simulator works as follows: first, it generates a set of content requests concentrated at day-time; inter-arrival times between successive requests are exponentially distributed according to the IRM model [8] which is the de facto standard in the analysis of storage systems. Next, the simulator associates to each request the coordinates (and the mobility according to the SLAW model) of the user requesting the content. Then, it allocates content in

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

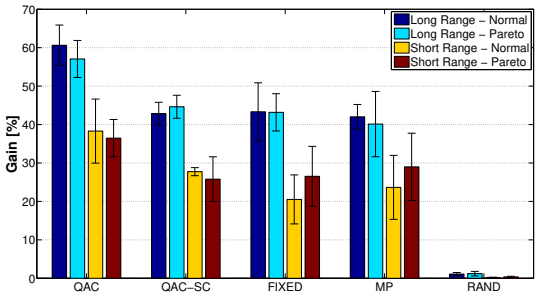



**Figure 3: Offloading gains.**

caches according to different allocation policies. For each request, a user downloads chunks of video when she is in the communication range of a vehicle storing the requested content. When the deadline expires, the potential remaining bytes are downloaded from the cellular infrastructure.

We consider and compare the following allocation policies:

- *QAC*. This policy solves the optimization problem with a reasonable approximation for content of generic size. This policy is described in Section 4.2.
- *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 suitable for content of small size and is described in Vigneri *et al.* [24].
- *QAC-SC*. This policy solves the optimization problem of *FIXED* when deadlines are variable. The problem is biconvex and is solved numerically.
- *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.
- *RAND*. Content is allocated randomly with fixed deadlines.

## 5.2 Caching policies evaluation

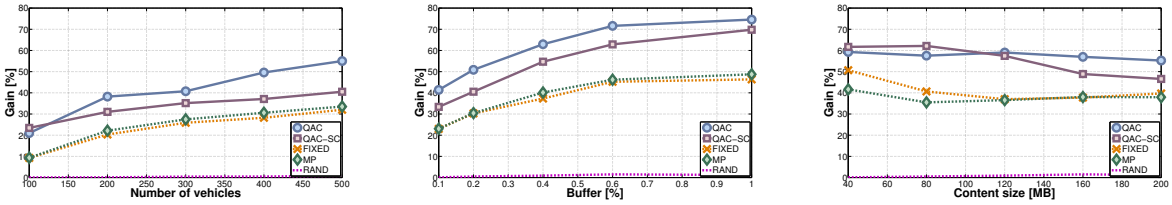
In Fig. 3 we plot the amount of data offloaded for different allocation policies. This plot also includes the 95% confidence interval. The fraction of traffic offloaded by *QAC* is much larger (additional gains of around 20%) than any other policy in any situation. For instance, when long range communications are considered, offloading gains are in the order of 60% for *QAC*, and no more than 40% for *QAC-SC*, *FIXED* and *MP*. *RAND* policy performs poorly in any scenario. It is also interesting to note that, while *QAC-SC* is expected to benefit from the deadline variability, it performs similar to fixed deadline policies since the assumption that a content can be downloaded in one contact is unrealistic for content of 200 MB. Not substantial differences have been observed for different 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.

Fig. 4 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. 4a 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 Lemma 4.3). What is more, from Corollary 4.8, the error of the approximation used by *QAC* is proportional to the standard deviation of $B_i$ which increases in a sparse environment.

Fig. 4b compares different buffer capacities per vehicle. Buffer size goes from the 0,1% to the 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 an entire Torrent catalogue (~3 PB) or the entire Netflix catalogue (~3 PB), 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). Furthermore, as the buffer 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. 4c we analyze the effect of content size by varying the mean content size from 30 MB to 200 MB. As expected, for small content, *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 uses the same assumption. What is important to notice, however, is that the traffic offloaded by *QAC* is stable for any content size.

Finally, we perform an analysis of the user QoE by allowing different values of $\omega_{max}$. In Fig. 5, we show the upper bound on the mean slowdown $\omega_{max}$ 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 200 MB, 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. 5 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 $\omega_{max}$ (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 variable deadlines. Knowing the function that ties user




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

**Figure 4: Fraction of traffic offloaded as a function of vehicle density (Fig. 4a), buffer capacity (Fig. 4b) and mean content size (Fig. 4c) for long range communications.**

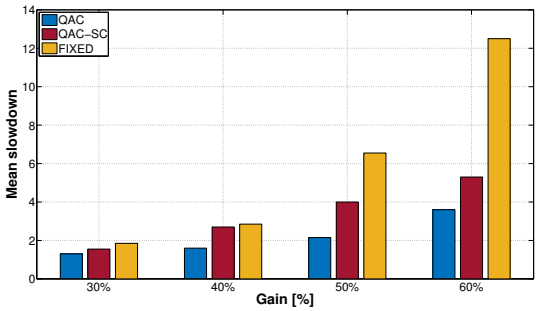



**Figure 5: Mean slowdown needed to reach specific offloading gains for long range communications.**

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.

## 6 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 deadlines by introducing a QoE metric; (iii) the generic model includes per chunk-level downloads from vehicles. In the paper, we propose a caching policy that can be adopted by MNOs. This policy has 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 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.

## ACKNOWLEDGMENT

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

## REFERENCES

[1] BMW Vehicular CrowdCell. https://goo.gl/FBzemq, 2016.
[2] Veniam. https://veniam.com/, 2017.
[3] N. Alliance. NGMN 5G White Paper. https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf, 2015.
[4] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting Mobile 3G Using WiFi. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 209–222. ACM, 2010.
[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
[6] H. Cai, I. Koprulu, and N. B. Shroff. Exploiting double opportunities for deadline based content propagation in wireless networks. In *Proceedings IEEE INFOCOM*, pages 764–772, April 2013.
[7] Cisco. Cisco visual networking index: Global mobile data traffic forecast update. 2016-2021.
[8] E. G. Coffman and P. J. Denning. *Operating systems theory*, volume 973. Prentice-Hall Englewood Cliffs, NJ, 1973.
[9] V. Conan, J. Leguay, and T. Friedman. Characterizing Pairwise Inter-contact Patterns in Delay Tolerant Networks. In *Proceedings of the 1st International Conference on Autonomic Computing and Communication Systems (Autonomics)*, pages 19:1–19:9. ICST, 2007.
[10] G. Gao, M. Xiao, J. Wu, K. Han, and L. Huang. Deadline-Sensitive Mobile Data Offloading via Opportunistic Communications. In *13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2016.
[11] N. Golrezaei, A. G. Dimakis, and A. F. Molisch. Wireless Device-to-Device Communications with Distributed Caching. *CoRR*, abs/1205.7044, 2012.
[12] 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 Communications Magazine*, 51(4):142–149, 2013.
[13] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. 2013.
[14] R. Kannan and C. L. Monma. On the computational complexity of integer programming problems. In *Optimization and Operations Research*. Springer, 1978.
[15] T. Karagiannis, J. Y. L. Boudec, and M. Vojnovic. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE Transactions on Mobile Computing*, (10):1377–1390, Oct 2010.
[16] S. Karlin and H. Taylor. *A First Course in Stochastic Processes*. Elsevier Science, 2012.
[17] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. SLAW: A New Mobility Model for Human Walks. In *IEEE INFOCOM*, pages 855–863, April 2009.
[18] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong. Mobile Data Offloading: How Much Can WiFi Deliver? *IEEE/ACM Transactions on Networking*, April 2013.
[19] 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*, pages 2364–2372, April 2014.
[20] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. DAD data set epfl/mobility (v. 2009-02-24). http://crawdad.org/epfl/mobility/, Feb 2009.
[21] K. Poularakis, G. Iosifidis, A. Argyriou, and L. Tassiulas. Video delivery over heterogeneous cellular networks: Optimizing cost and performance. In *IEEE INFOCOM Conference on Computer Communications*, April 2014.
[22] P. Raghavan and C. D. Tompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, 1987.
[23] H. Schmidli. Lecture Notes on Risk Theory.
[24] L. Vigneri, T. Spyropoulos, and C. Barakat. Storage on wheels: Offloading popular contents through a vehicular cloud. In *IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016.
[25] M. Zeni, D. Miorandi, and F. De Pellegrini. YOUStatAnalyzer: a tool for analysing the dynamics of YouTube content popularity. In *Proc. 7th International Conference on Performance Evaluation Methodologies and Tools*, Torino, Italy, 2013.