Per-Chunk Caching for Video Streaming from a Vehicular Cloud

Luigi Vigneri
EURECOM
Biot, France 06410
luigi.vigneri@eurecom.fr

Salvatore Pecoraro
EURECOM
Biot, France 06410
salvatore.pecoraro@eurecom.fr

Thrasyvoulos Spyropoulos
EURECOM
Biot, France 06410
thrasyvoulos.spyropoulos@eurecom.fr

Chadi Barakat
INRIA Sophia Antipolis
Valbonne, France 06902
chadi.barakat@inria.fr

ABSTRACT

Caching content at the edge of mobile networks is considered as a promising way to deal with the data tsunami. In addition to caching at fixed base stations or user devices, it has been recently proposed that an architecture with public or private transportation acting as mobile relays and caches might be a promising middle ground. In previous work, we have assumed users are streaming video files and have analyzed how many replicas of each video file to cache in such a vehicular fleet working towards minimizing the amount of bits per file downloaded from (expensive) infrastructure links. However, this work has been assuming that a vehicle will store the entire content, or none of it. In practice, later chunks have an inherent "delay tolerance" as there is more time to find them before they must be played out. What is more, numerous studies as well as everyday experience suggest that most files (e.g., YouTube) are not entirely watched. This makes the previous policies suboptimal, as fewer (or no) replicas could be allocated to late chunks of a file and more to the most popular chunks. In this work, we formulate an optimization problem to compute the optimal allocation *per-chunk*, to minimize the load on the cellular infrastructure, and we show that significant performance gains can be achieved compared to *per-content* allocation policies.

1 INTRODUCTION

The recent diffusion of handheld devices is driving an exponential increase of the mobile traffic demand which is already overloading the cellular infrastructure [2]. Mobile network operators are trying to keep up by deploying small cells in urban environments and by edge caching (at femtocells, Wi-Fi access points or even user equipments) [6, 13]. Recently, both industry [1] and academia [17] have proposed to use vehicles acting as mobile relays to store replicas of popular content. Thanks to their intrinsic mobility, vehicles can increase the "effective" storage capacity a user has access to (since a user can meet a larger number of vehicles, compared to fixed small cells, in the same time interval [17]) and also have favorable characteristics in terms of CAPEX/OPEX compared to fixed outdoors small cells.

In our previous studies, we have considered the problem of downloading whole content [17] or streaming videos [18] from vehicles acting as mobile caches (*vehicular cloud*). Intuitively, the rough tradeoff is the following: the higher the number of vehicles caching a content, the higher the chance that this content (or chunk) will be downloaded from such a vehicle "on time" (rather than from expensive macro-cell links); however, the marginal benefit of each extra replica is decreasing, creating a non-trivial tradeoff between content popularity and caching policy. In our latter work [18], we have further assumed that content is streamed chunk-by-chunk, and if a chunk (e.g., corresponding to the 30th minute of a video) can be downloaded from an encountered vehicle before it must be played (i.e., up to 30 minutes after the user starts watching the content) this is data traffic that is offloaded from the main infrastructure *without any visible impact on the user.*

However, in that work we have made the simplifying assumption that either all chunks or no chunks of a content must be cached in a vehicle. Said otherwise, every chunk of a content must have the same number of replicas. Yet, as evident by the above example, early chunks have a smaller chance to be downloaded from a vehicle than later chunks, due to the larger inherent delay tolerance of the latter. At the same time, common experience as well as recent measurements suggest that a lot of video content (e.g., YouTube files) are only partially watched as users often abandon the playout of a video before it ends. These two opposing "forces" call for a per-chunk optimization policy: (i) the former suggests that it is perhaps wasteful to cache too many of the early chunks and give instead more space to later ones that have a higher chance to be offloaded; (ii) the latter suggests that early chunks have different popularity than later chunks and thus perhaps deserve more storage space. Our goal in this paper is to address this tradeoff and propose per-chunk cache allocation policies that outperform existing ones. To our best knowledge, this is the first work to consider per-chunk caching in such a vehicular context. We make the following contributions:

- *Modelling.* We model the video streaming of content[1] when chunks can be opportunistically downloaded from nearby vehicles. In Section 2 we state the main assumptions, and we formulate an optimization problem to compute an allocation

CHANTS'17, October 20, 2017, Snowbird, UT, USA.
© 2017 ACM. ISBN 978-1-4503-5144-7/17/10…$15.00
DOI: https://doi.org/10.1145/3124087.3124095

[1] Note that this scenario does not include live content streaming which is not usually amenable to caching, and is often optimized using multicast techniques.

(per-chunk) maximizing the expected amount of offloaded traffic as a function of network characteristics (e.g., vehicle density, chunk popularity).

- *Optimization.* In Section 3, based on this model, we propose two appropriate approximations according to the download rate from vehicles. We show that these problems are NP-hard, and we solve a related continuous relaxation.
- *Performance analysis.* In Section 4 we validate our theoretical results using real traces for content popularity and vehicle mobility, and show that our system can offload up to 45% of streamed data in realistic scenarios even with modest technology penetration, which is more than 10% larger than traditional content caching techniques.

Finally, we conclude our paper in Section 5 with a summary and future work.

2 SYSTEM MODEL

We introduce here the system model and related assumptions.

2.1 Video Streaming Model

We consider a network with three types of nodes:

- *Infrastructure nodes* ($\mathcal{I}$). Base stations or macro-cells. They provide full coverage and can serve any content request.
- *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 (non-live) video content for streaming to $\mathcal{H}$ and $\mathcal{I}$ nodes.

Each video consists of a number of small chunks that are downloaded into a $\mathcal{U}$ node's playout buffer in order and consumed for playout as follows:

- *Helper download.* When a $\mathcal{U}$ node is in range of an $\mathcal{H}$ node that stores the requested chunks, the next chunks not yet in the playout buffer are downloaded at low cost. The number of chunks that can be downloaded during a contact depends on the download rate, the contact duration, etc.
- *Infrastructure download.* When a $\mathcal{U}$ node is not in range of an $\mathcal{H}$ node that stores the requested content *and* its playout buffer is (almost) empty, new chunks are downloaded from the infrastructure at a mean rate r_I until another $\mathcal{H}$ node storing the content is encountered. The communication between $\mathcal{U}$ and $\mathcal{I}$ nodes has a high cost in terms of energy consumption and bandwidth of the backhaul links [4].
- *Playout.* Chunks in the playout buffer are consumed at a mean viewing *playout* rate r_P. For simplicity, we assume that each chunk corresponds to a fixed video duration of τ seconds, and has the same size in terms of bytes[2].

2.2 Main Assumptions

A.1 - Catalogue. Let $\mathcal{K}$ be the set of all possible contents that users might request (also defined as "catalogue") where $|\mathcal{K}| = k$. A content $i \in \mathcal{K}$ is divided into s chunks of equal size. Each chunk is characterized by a popularity value ϕ_{ij} 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, 13], we assume this time window to be a system parameter chosen by the cellular operator. Every time window, the cellular operator refreshes its caches installed in vehicles according to the new estimated popularity[3].

A.2 - Mobility model. We assume that the inter-meeting times T_{ij} between a user requesting chunk j of content $i \in \mathcal{K}$ and *any* vehicle storing such a chunk are independent and identically distributed random variables characterized by an exponential distribution[4] with mean rate λ. Exponential inter-meeting times have been largely used in literature and considered as a good approximation, especially in the tail of the distribution [3, 8].

A.3 - Cache model. Let $x_{ij}^{(w)} \in \{0, 1\}$, $i \in \mathcal{K}, j \in \{0, \ldots, s\}$, $w \in \mathcal{H}$ be an indicator variable denoting if helper node w stores chunk j of content i. Let further x_i denote the number of $\mathcal{H}$ nodes storing content i:

$$x_{ij} = \sum_{w \in \mathcal{H}} x_{ij}^{(w)}.$$

The matrix $\mathbf{x}$ will be the control variable for our optimal cache allocation problem.

A.4 - Content download rate. We assume $r_I = r_P + \epsilon$ ($\epsilon > 0$ small) in order to limit the access to the cellular infrastructure to the minimum required to ensure smooth playout (for simplicity, we assume ϵ equal to 0). We also assume that the download from the helper nodes is faster than r_I, i.e., $r_H \geq r_I > r_P$. These are reasonable assumptions due to the reduced communication distance: scenarios where r_I (and/or r_H) are lower than the playout rate require initial buffering which is known to significantly degrade QoE [7], and are orthogonal issues to the problem addressed in this paper.

A.5 - Data offloading. A request for content i will download a number of chunks from $\mathcal{H}$ nodes. This number is a random variable that depends on x_{ij} as well as random mobility variables (intercontact times and contact durations).

The notation used in the paper is summarized in Table 1.

2.3 Problem formulation

Given the above assumptions, we can propose a policy where: (i) the user's video is never interrupted provided the infrastructure can guarantee at least the playout rate (if that is not the case, then this is an issue of the infrastructure); (ii) while the user is watching the video, future parts of it are actually downloaded from locally encountered caches (in principle pre-fetched) thus offloading some traffic from the infrastructure. As long as the playout buffer remains non-empty, $\mathcal{I}$ nodes never need to be accessed. And when they do, we ensure that the minimum necessary amount of bytes is downloaded from the infrastructure ($r_I = r_P + \epsilon$). The goal of the paper is to find the allocation that minimizes the number chunks

[2] This is in general not the case as the actual size depends on the compression factor, type of frame, etc. However, we are interested first in an average case analysis and such differences cancel themselves out when considering enough chunks.

[3] Several studies have confirmed that simple statistical models (e.g., ARMA models) along with content type characteristics can help to have good estimation of the request rate, at least in the immediate future [9, 16].

[4] In case of heterogeneous mobility, λ could be seen as the average among the various λ_{ij}, which works well as a first order approximation [15].

Table 1: Notation used in the paper.

CONTROL VARIABLES	
x_i	Number of replicas stored for content i
CONTENT	
k	Number of content in the catalogue
ϕ_{ij}	Request rate for chunk j of content i
s_i	Number of chunks of content i
c	Buffer size per vehicle
MOBILITY	
λ	Mean inter-meeting rate between $\mathcal{U}$ and $\mathcal{H}$ nodes
h	Number of vehicles
CHUNK DOWNLOAD	
r_H	Mean download rate from $\mathcal{H}$ nodes
r_I	Mean download rate from $\mathcal{I}$ nodes (equal to r_P)

downloaded from the cellular infrastructure needed to ensure uninterrupted streaming. Thus, we formulate the following optimization problem according to the previous assumptions:

Problem 1. *The solution to the following optimization problem maximizes the expected number of chunks offloaded through the vehicular cloud:*

$$\underset{\mathbf{x}\in X^{k\times s}}{\text{maximize}} \quad \sum_{i=1}^{k}\sum_{j=1}^{s}\phi_{ij}\cdot\int_{0}^{(j-1)\cdot\tau}\mathbf{P}[A_{ij}\mid\mathbf{x},t]\,dt, \tag{1}$$

$$\text{subject to} \quad \sum_{i=1}^{k}\sum_{j=1}^{s}x_{ij}^{(w)}\le c, \quad \forall w\in\mathcal{H}, \tag{2}$$

where A_{ij} is the event that chunk j of content i is offloaded through the vehicular cloud, $\mathbf{P}[A_{ij}\mid\mathbf{x},t]$ is the probability of downloading chunk j of video i at time t, given caching allocation $\mathbf{x}$, and $X\triangleq\{a\in\mathbb{N}\mid 0\le a\le h\}$ is the feasible region for the control variable $\mathbf{x}$.

The objective function counts the number of chunks downloaded from the cellular infrastructure in a seeding time window for the entire catalogue. For each content, this is equivalent to the content popularity times the probability to download a chunk before it must be played out. Since each chunk has a duration of τ seconds this time is equal to $(j-1)\cdot\tau$ (for the j-th chunk of a content). The number of replicas is bounded by the number of vehicles participating in the vehicular cloud. What is more, each vehicle has a storage constraint and cannot store more than c bytes (see Eq. (2)).

3 ANALYTICAL MODEL

Problem (1) is hard to solve for a number of reasons: it is basically a discrete optimization problem, and the probability to download a given chunk depends on the probability to download past chunks and is hard to derive analytically, in general. Instead, we propose two approximate ways to solve the problem, namely for infinite (Section 3.1) and limited contact bandwidth (Section 3.2) (between users and vehicles).

3.1 Infinite bandwidth

We introduce the following definition:

Definition 3.1 (Infinite bandwidth). Assume that a user is streaming content i. If a vehicle storing a subset of chunks for content i comes in range to the user, in the *infinite bandwidth* scenario we assume the user is able to download *all* these chunks.

Note, we *do not* require that this is true in a real setup. We simply say that *our policy* will assume so, for simplicity (and thus might not always be optimal). This assumption becomes more accurate if for example there are many vehicles and many contents to ensure each one stores only few chunks per content[5], and of course depends also on the content size and download rate between $\mathcal{H}$ and $\mathcal{I}$ nodes. According to Definition 3.1, Problem (1) can be rewritten as follows:

Problem 2. *Consider the infinite bandwidth scenario. The solution to the following optimization problem maximizes the expected number of bytes offloaded through the vehicular cloud:*

$$\underset{\mathbf{x}\in X^{k\times s}}{\text{maximize}} \quad \sum_{i=1}^{k}\sum_{j=1}^{s}\phi_{ij}\cdot(1-e^{-\lambda\cdot(j-1)\cdot\tau\cdot x_{ij}}),$$

$$\text{subject to} \quad \sum_{i=1}^{k}\sum_{j=1}^{s}x_{ij}^{(w)}\le c, \quad \forall w\in\mathcal{H}. \tag{3}$$

Proof. When there is infinite bandwidth per contact

$$\mathbf{P}[A_{ij}\mid\mathbf{x},t]=\mathbf{P}[A_{ij}\mid\mathbf{x}].$$

Assuming further exponential inter-contact times, the probability to find a vehicle storing chunk ij before its playout time is

$$\int_{0}^{(j-1)\cdot\tau}\mathbf{P}[A_{ij}\mid\mathbf{x}]=\mathbf{P}[T_{ij}<(j-1)\cdot\tau]$$
$$=1-e^{-\lambda\cdot(j-1)\cdot\tau\cdot x_{ij}},$$

where the second equivalence is true because T_{ij} follows an exponential distribution with rate λ times x_{ij} replicas for the given chunk. The constraints simply ensure the capacity of each vehicle is not violated. □

Problem (2) is an NP-hard combinatorial problem, when the allocation variables x_{ij} are integer, as it is a bounded knapsack problem (BKP) with a nonlinear objective function [11]. A standard approach in such cases is to consider a *continuous relaxation* of the problem. This not only converts the problem to a convex one (as we will see), that can be solved efficiently, but also can be solved analytically, giving valuable insights into the optimal allocation. Furthermore, we can replace the *individual* capacity constraint of Eq. (2) with a *global* capacity constraint, i.e.,

$$\sum_{i=1}^{k}\sum_{j=1}^{s}x_{ij}\le c\cdot h. \tag{4}$$

It is easy to see that, if $\mathbf{x}$ is fractional, any allocation that fits the global capacity is also a feasible allocation, given the assumption of IID mobility[6]. We can prove the following result:

[5] The specific allocation of chunks per vehicle is a different problem that is left as a future work.

[6] We define *feasible* an allocation that satisfies the individual capacity constraint of Eq. (2).

Theorem 3.2. *The solution of Problem (2) when* $\mathbf{x}$ *is real is given by*

$$x_{ij} = \begin{cases} 0, & \textit{if } \phi_{ij} < L, \\ \frac{1}{\lambda \cdot (j-1) \cdot \tau} \cdot \ln\left(\frac{\lambda \cdot (j-1) \cdot \tau \cdot \phi_{ij}}{\rho}\right), & \textit{if } L \leq \phi_{ij} \leq U, \\ h, & \textit{if } \phi_i > U, \end{cases}$$

where ρ *is an appropriate Lagrange multiplier,* $L \triangleq \frac{\rho}{\lambda \cdot (j-1) \cdot \tau}$, $U \triangleq \frac{\rho \cdot e^{h \cdot \lambda \cdot (j-1) \cdot \tau}}{\lambda \cdot (j-1) \cdot \tau}$.

Proof. Problem (2) is a convex optimization problem since its objective function is convex (because it is the sum of convex functions), the constraint is linear and the set of feasible solutions is convex. We solve it by Karush-Kuhn-Tucker (KKT) conditions. For such a convex problem, this method provides necessary and sufficient conditions for the stationary points to be optimal solutions. The KKT conditions for Problem (2) are

$$\begin{cases} l_{ij} \cdot x_{ij} = 0 \\ m_{ij} \cdot (h - x_{ij}) = 0 \\ \rho \cdot \left(c \cdot h - \sum_{i=1}^{k} \sum_{j=1}^{s} x_{ij}\right) = 0 \end{cases}$$

where l_{ij} and m_{ij} are appropriate Lagrange multipliers related to the bounds of $\mathbf{x}$. The related Lagrangian function $\mathcal{L}(\mathbf{x})$ is

$$\mathcal{L}(\mathbf{x}) = \sum_{i=1}^{k} \sum_{j=1}^{s} \left[\phi_{ij} \cdot (1 - e^{-\lambda \cdot (j-1) \cdot \tau \cdot x_{ij}}) + l_{ij} \cdot x_{ij}\right] + $$

$$+ \sum_{i=1}^{k} \sum_{j=1}^{s} m_{ij} \cdot (h - x_{ij}) + \rho \cdot \left(c \cdot h - \sum_{i=1}^{k} \sum_{j=1}^{s} x_{ij}\right).$$

We compute the stationary points by computing the derivative of the Lagrangian function for each content i. Since the problem is convex, these points are also global solutions.

$$\frac{d\mathcal{L}(\mathbf{x})}{dx_{ij}} = \lambda \cdot (j-1) \cdot \tau \cdot \phi_{ij} \cdot e^{-\lambda \cdot (j-1) \cdot \tau \cdot x_{ij}} + l_{ij} - m_{ij} - \rho = 0.$$

Making explicit $\mathbf{x}$, we obtain:

$$x_{ij} = \frac{1}{\lambda \cdot (j-1) \cdot \tau} \cdot \ln\left(\frac{\lambda \cdot (j-1) \cdot \tau \cdot \phi_{ij}}{\rho - l_{ij} + m_{ij}}\right).$$

Then, the system constraints create three regimes depending on the content popularity:

- *Low popularity.* The optimal allocation $\mathbf{x}$ must be greater or equal to 0. According to the KKT conditions, we have two cases that satisfy the constraint: (i) $x_{ij} > 0,\ l_{ij} = 0$; (ii) $x_{ij} = 0,\ l_{ij} > 0$. The threshold between case (i) and (ii) depends on the content popularity: specifically, a content will get more than 0 copies when its popularity is higher than L which can be easily computed when $x_{ij} > 0$:

$$\frac{1}{\lambda \cdot (j-1) \cdot \tau} \cdot \ln\left(\frac{\lambda \cdot (j-1) \cdot \tau \cdot \phi_{ij}}{\rho}\right) > 0$$

$$\phi_{ij} > \frac{\rho}{\lambda \cdot (j-1) \cdot \tau} \triangleq L.$$

- *High popularity.* The content allocation is upper bounded by the number vehicles h participating in the cloud. Similarly to the previous scenario, due to the KKT conditions, the constraint is satisfied when: (i) $x_i < h,\ m_i = 0$; (ii) $x_i = h,\ m_i > 0$:

$$\phi_{ij} < \frac{\rho \cdot e^{h \cdot \lambda \cdot (j-1) \cdot \tau}}{\lambda \cdot (j-1) \cdot \tau} \triangleq U.$$

- *Medium popularity.* In all the other cases (i.e., when $U \leq \phi_{ij} \leq L$), the optimal allocation is proportional to the logarithm of the content popularity.

□

As a final step, we use *randomized rounding* [14] to go back to an integer allocation, which is a widely used approach for designing and analyzing such approximation algorithms. Furthermore, when cache sizes are large enough, such methods are expected to introduce very small approximation errors.

3.2 Limited bandwidth

The infinite bandwidth model provides some initial insights on a per-chunk allocation for content streaming. In reality, contact duration (or equivalently the available bandwidth between $\mathcal{U}$ and $\mathcal{H}$) is limited and only a few chunks can be downloaded per contact. In this subsection, we describe a model that exploits such a limited bandwidth to improve the previous problem formulation. We introduce the following definition:

Definition 3.3 (Limited bandwidth). Assume that a given user requests content i. Assume further that the playout of such a content has started at time 0, and a vehicle is met at time t. In the limited bandwidth scenario, the user is able to download a chunk j of content i with probability $p_{ij}(z)$ during a contact, where $z \triangleq j \cdot \tau - t$ is the residual playout time, and $p_{ij}(z)$ is non-decreasing in z.

As an example, assume that a content i is split in 10 chunks which are stored in all vehicles. Assume that, at time 0 (i.e. when the user starts watching), a user is in the communication range of a vehicle storing chunks 1 to 5. Since chunks are downloaded in order (this is the optimal policy to avoid missing deadlines), it is easy to see that the probability to download chunk 1 ($p_{i1}(0)$) is higher than the probability to download chunk 5, as the contact with that vehicle might end before all chunks before 5 are downloaded. However, if the user is currently watching chunk 3 and encounters the same vehicle, only chunks 4 and 5 need to be downloaded, hence increasing the chances of getting 5 before the contact is over.

Probability $p_{ij}(z)$ is very hard to model analytically, because it depends on vehicle mobility, allocation of earlier chunks to vehicles and residual playout time z. To simplify things, we can assume as a first step that this probability is independent of the allocation, but is a *given* (non-decreasing) function of time (this function could be estimated, for example, as the mean over a number of contact sample paths, for a given allocation). In that case, we can write the following optimization problem:

Problem 3. *Consider the limited bandwidth scenario. The solution to the following optimization problem maximizes the expected number* *of bytes offloaded through the vehicular cloud:*

$$\begin{aligned}
&\underset{\mathbf{x}\in X^{k\times s}}{\mathit{maximize}} && \sum_{i=1}^{k}\sum_{j=1}^{s}\phi_{ij}\cdot(1-e^{-Q_{ij}\cdot\lambda\cdot(j-1)\cdot\tau\cdot x_{ij}}),\\
&\mathit{subject\ to} && \sum_{i=1}^{k}\sum_{j=1}^{s}x_{ij}^{(w)}\leq c,\quad \forall w\in\mathcal{H}, \qquad (5)
\end{aligned}$$

where $Q_{ij} \triangleq \frac{1}{(j-1)\cdot\tau}\cdot\int_0^{(j-1)\cdot\tau} p_{ij}(z)\,dz$.

Proof. In the *limited bandwidth* model, a chunk can be downloaded with probability $p_{ij}(z) \leq 1$ during a meeting. If the contacts with vehicles storing the requested content follow a Poisson process having mean rate $\lambda \cdot x_{ij}$, as we have assumed, $p_{ij}(z)$ introduces a *thinning* of the original Poisson process and leads to a Poisson process with rate $p_{ij}(t) \cdot \lambda \cdot x_{ij}$. The latter is the rate of *successful* contacts (i.e., contacts *with* vehicles storing chunk *ij*, and which *are* long enough to download chunk *ij*. We can further make a *mean value* approximation and replace this time-dependent probability with its mean over the entire playout interval:

$$Q_{ij} \triangleq \frac{1}{(j-1)\cdot\tau}\cdot\int_0^{(j-1)\cdot\tau} p_{ij}(z)\,dz$$

This basically introduces one "thinning" parameter for each content chunk, Q_{ij}. The rest of the proof then continues as in the case of Problem (2). □

The following result then holds, in terms of the optimal allocation for this limited bandwidth case. The proof follows that of Theorem 3.2 replacing λ with $Q_{ij}\lambda$, and is omitted.

Theorem 3.4. *The solution of Problem (2) when* $\mathbf{x}$ *is real is given by*

$$x_{ij} = \begin{cases} 0, & \mathit{if}\,\phi_{ij} < L,\\ \frac{1}{Q_{ij}\cdot\lambda\cdot(j-1)\cdot\tau}\cdot\ln\left(\frac{Q_{ij}\cdot\lambda\cdot(j-1)\cdot\tau\cdot\phi_{ij}}{\rho}\right), & \mathit{if}\,L \leq \phi_{ij} \leq U,\\ h, & \mathit{if}\,\phi_i > U,\end{cases}$$

where $L \triangleq \frac{\rho}{Q_{ij}\cdot\lambda\cdot(j-1)\cdot\tau}$ *and* $U \triangleq \frac{\rho\cdot e^{h\cdot Q_{ij}\cdot\lambda\cdot(j-1)\cdot\tau}}{Q_{ij}\cdot\lambda\cdot(j-1)\cdot\tau}$.

As is evident by the above equations, the optimal allocation is now dependent on another chunk-dependent parameter, Q_{ij}.

4 SIMULATIONS

4.1 Simulation setup

To validate our results, we perform simulations based on real traces for (i) mobility and (ii) content popularity:

- **Mobility**: We use the Cabspotting trace [12] to simulate the vehicle behaviour; this trace records the GPS coordinates for 531 taxis in San Francisco for more than 3 weeks with granularity of 1 minute. In order to improve the accuracy of our simulations, we increase the granularity to 10 seconds by linear interpolation. We extrapolate the mobility statistics (e.g., λ) from the analysis of the Cabspotting trace to compute the optimal allocation.
- **Content popularity**: We infer the number of requests per day from a database with statistics for 100.000 YouTube videos [19]. The database includes static (e.g., title, description, author, duration) and dynamic information (e.g., daily and cumulative views, shares, comments). Data is worldwide, and we scale it linearly according to the estimated population of the centre of San Francisco. We assume that each content is split into 10 chunks of equal size, and that the internal chunk popularity follows a Zipf-like distribution [10].

We build a MATLAB simulator as follows: first, we generate a set of requests, and we associate a random location (in GPS coordinates) to each one. The number of requests per content per day is given by the YouTube trace. Then, we store chunks in vehicles according to chosen allocation policy. For each request, we simulate the playout of the video; the end user buffer will be opportunistically filled when the vehicular cloud can be contacted, according to the mobility provided by the Cabspotting trace. The number of chunks downloaded per contact depends on the contact duration and on the distance between user and vehicle. In our simulations, we assume that end users can contact the vehicular cloud with either *short* (100 m) or *long* (200 m) range communications. The user abandons the playout of the video with some probability: specifically, when a user watches a chunk, she decides to watch the following chunk with probability w or to abandon the playout with probability $1 - w$. In order to increase the number of simulations and to provide sensitivity analysis for content size and buffer capacity, we limit the number of content to 10.000. We scale down the vehicle storage capacity c to ensure that 0,1% of the total catalogue fits in each cache (i.e., 10 entire contents).

We compare the following allocation policies:

- *Per Chunk Caching (PCC).* This policy allocates chunks of popular content according to the model described in Section 3.1.
- *Generic Traffic (GT).* This policy allocates videos optimally according to the Generic Traffic policy described in Section 3.2 of Vigneri *et al.* [18].
- *Low Traffic (LT).* This policy allocates videos optimally according to the Low Traffic policy described in Section 3.1 of Vigneri *et al.* [18].

Our main goal in this preliminary evaluation is to highlight the improvements, in terms of number of chunks offloaded, brought by caching *per chunk* compared to *per content*, in a relatively realistic scenario. We therefore focus on the simpler unlimited bandwidth policy. Once more, we stress that *the actual contacts in the simulator do not have unlimited bandwidth, but only that our allocation policy is designed making this simplifying assumption.* Nevertheless, even this suboptimal policy already brings considerable gains. Taking the limited bandwidth explicitly into account can provide additional gains, but requires a careful tuning of the $p_{ij}(z)$ function and is deferred to future work.

4.2 Performance evaluation

In Figure 1 we plot the percentage of chunks offloaded for different allocation policies when communication range is 100 m or 200 m. Specifically, when fractional storage is allowed (PCC), 10% more traffic can be offloaded (from 35% to 45%), a relative improvent of around 30%. As already mentioned, this happens because later chunks have less probability to be watched which is not taken into account by *LT* and *GT* policies. Rather, *PCC* replaces some of these chunks, that are not likely to be watched, with some others. Such a replacement brings interesting offloading gains. What is more, the

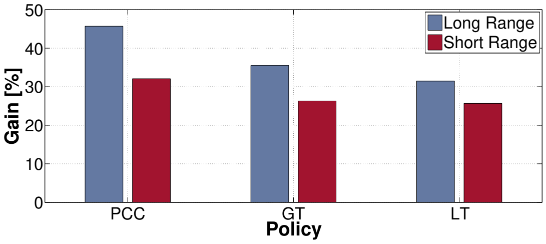

Figure 1: Percentage of chunks offloaded through the vehicular cloud for short and long range communications.

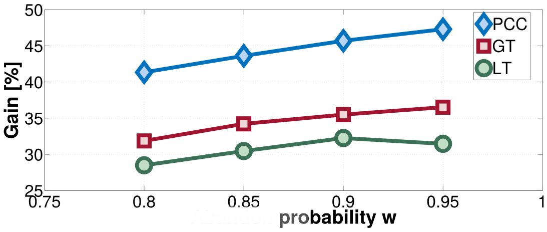

Figure 2: Percentage of chunks offloaded through the vehicular cloud according to the probability of watching the subsequent chunk.

way content is allocated into vehicles (chunks of the same content in the same vehicle vs. chunks of the same content spread over all vehicles) can largely affect the amount of data offloaded, and can be used by an operator to further increase performance. Because of the large number of requests in the period considered, the confidence interval is too small to be distinguishable and hence is ignored.

In Figure 2 we perform sensitivity analysis according to the probability of viewing the subsequent chunk when communication range is 200 m. We analyse the range 0, 80 – 0, 95 (i.e., a video is entirely played out from 10 to 60 % of times). Similarly to the previous scenario, *PCC* provides a relative improvement of at least 30% (or 10% in absolute values) in all scenarios. What is more, the efficiency of the policy improves as the abandonment probability increases. Although this confirms the effectiveness of our proposal, we believe that a finer per chunk policy (e.g., limited bandwidth with an appropriate function $p_{ij}(t)$) would provide even larger gains. Interestingly, considerable gains are 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 % of the catalogue. E.g., if one considers an entire Torrent or Netflix catalogue (~3 PB), a vehicle cache of about 3 TB already suffices to offload more than 40 % of the total traffic for long range communications (and 30%for short range).

5 CONCLUSION

In this paper, we have introduced a per chunk allocation policy for video streaming from a vehicular cloud. We have shown that the related optimization problem is hard, and we have solved two reasonable approximations that make different assumptions about the number of chunks that can be downloaded per contact. The simulations performed have confirmed that caching policies that differentiate between chunks can offload much more traffic, compare to policies that treat all chunks of the same content the same. A number of interesting problems remain open, such as the proper modeling of the per chunk download success probability, as well as the specific allocation of chunks to vehicles, given that the number of replicas needed are known.

ACKNOWLEDGEMENT

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] 2017. Veniam. https://veniam.com. (2017).

[2] Cisco. 2016-2021. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update.

[3] Vania Conan et al. 2007. Characterizing Pairwise Inter-contact Patterns in Delay Tolerant Networks *(Autonomics)*.

[4] Small Cell Forum. 2013. Backhaul Technologies for Small Cells: Use Cases, Requirements and Solutions. (Feb 2013).

[5] Negin Golrezaei, Andreas F. Molisch, Alexandros G. Dimakis, and Giuseppe Caire. 2012. Femtocaching and Device-to-Device Collaboration: A New Architecture for Wireless Video Distribution. *CoRR* abs/1204.1595 (2012). http://arxiv.org/abs/1204.1595

[6] Negin Golrezaei et al. 2012. Wireless Device-to-Device Communications with Distributed Caching. *CoRR* abs/1205.7044 (2012). http://arxiv.org/abs/1205.7044

[7] T. Hossfeld et al. 2012. Initial delay vs. interruptions: Between the devil and the deep blue sea. In *W. on QoMEX*.

[8] T. Karagiannis et al. 2010. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE Trans. on Mobile Computing* (2010).

[9] J. G. Lee, S. Moon, and K. Salamatian. 2010. An Approach to Model and Predict the Popularity of Online Contents with Explanatory Factors. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1. 623–630. DOI:http://dx.doi.org/10.1109/WI-IAT.2010.209

[10] S. H. Lim, Y. B. Ko, G. H. Jung, J. Kim, and M. W. Jang. 2014. Inter-Chunk Popularity-Based Edge-First Caching in Content-Centric Networking. *IEEE Communications Letters* 18, 8 (Aug 2014), 1331–1334. DOI:http://dx.doi.org/10.1109/LCOMM.2014.2329482

[11] Silvano Martello and Paolo Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, Inc., New York, NY, USA.

[12] M. Piorkowski et al. 2009. DAD data set epfl/mobility (v. 2009-02-24). http://crawdad.org/epfl/mobility/. (2009).

[13] K. Poularakis et al. 2014. Video delivery over heterogeneous cellular networks: Optimizing cost and performance. In *IEEE INFOCOM*. DOI:http://dx.doi.org/10.1109/INFOCOM.2014.6848038

[14] Prabhakar Raghavan and Clark D. Tompson. 1987. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica* 7, 4 (1987), 365–374. DOI:http://dx.doi.org/10.1007/BF02579324

[15] Pavlos Sermpezis and Thrasyvoulos Spyropoulos. 2016. Delay Analysis of Epidemic Schemes in Sparse and Dense Heterogeneous Contact Networks. *IEEE Transactions on Mobile Computing* (2016).

[16] G. Szabo and B.A. Huberman. 2010. Predicting the popularity of online content. *Comm. of the ACM* (2010).

[17] Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. 2016. 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)*.

[18] Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. 2016. Streaming Content from a Vehicular Cloud. In *Proceedings of the Eleventh ACM Workshop on Challenged Networks (CHANTS ’16)*. ACM, New York, NY, USA, 39–44. DOI:http://dx.doi.org/10.1145/2979683.2979684

[19] Mattia Zeni, Daniele Miorandi, and Francesco De Pellegrini. 2013. YOUStatAnalyzer: a Tool for Analysing the Dynamics of YouTube Content Popularity. In *Proc. 7th International Conference on Performance Evaluation Methodologies and Tools (Valuetools, Torino, Italy, December 2013)*. Torino, Italy.