UNIVERSITÉ CÔTE D'AZUR

École doctorale ED STIC
Unité de recherche: EURECOM/INRIA

Thèse de doctorat

Présentée en vue de l'obtention du
grade de docteur en *Computer Science* de
l'UNIVERSITE COTE D'AZUR

par
Luigi Vigneri

Les Véhicules comme un Mobile Cloud: Modélisation, Optimisation et Analyse des Performances

Dirigée par *Chadi Barakat* et
co-encadrée par *Thrasyvoulos Spyropoulos*

Soutenue le 11 juillet 2017
Devant le jury composé de

Chadi Barakat	Chercheur (HDR), *INRIA*	Directeur de thèse
Petros Elia	Professeur (HDR), *EURECOM*	Examinateur
Iordanis Koutsopoulos	Professeur associé, *AUEB*	Rapporteur
Emilio Leonardi	Professeur, *Politecnico di Torino*	Rapporteur
Giovanni Neglia	Chercheur (HDR), *INRIA*	Examinateur
Katia Obraczka	Professeur, *University of California Santa Cruz*	Examinateur
Georgios Paschos	Ingenieur de recherche, *Huawei Technologies*	Examinateur
Thrasyvoulos Spyropoulos	Professeur assistant, *EURECOM*	Directeur de thèse

École Doctorale STIC

Doctoral Thesis

Vehicles as a Mobile Cloud: Modelling, Optimization and Performance Analysis

Author:
Luigi Vigneri

Supervisors:
Thrasyvoulos Spyropoulos
Chadi Barakat

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy from

Université Côte d'Azur

Declaration of Authorship

I, Luigi VIGNERI, declare that this thesis titled, "Vehicles as a Mobile Cloud: Modelling, Optimization and Performance Analysis" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Imagine a captain of a ship the moment a shift of direction must be made; then he may be able to say: I can do either this or that. But if he is not a mediocre captain he will also be aware that during all this the ship is ploughing ahead with its ordinary velocity, and thus there is but a single moment when it is inconsequential whether he does this or does that. So also with a person - if he forgets to take into account the velocity - there eventually comes a moment where it is no longer a matter of an Either/Or, not because he has chosen, but because he has refrained from it, which also can be expressed by saying: Because others have chosen for him-or because he has lost himself. "

Søren Aabye Kierkegaard

"Non è che la vita vada come tu te la immagini. Fa la sua strada. E tu la tua. E non sono la stessa strada. Io non è che volevo essere felice, questo no. Volevo... salvarmi, ecco: salvarmi. Ma ho capito tardi da che parte bisognava andare: dalla parte dei desideri. Uno si aspetta che siano altre cose a salvare la gente: il dovere, l'onestà, essere buoni, essere giusti. No. Sono i desideri che salvano. Sono l'unica cosa vera. Tu stai con loro, e ti salverai."

Alessandro Baricco

Abstract

Vehicles as a Mobile Cloud: Modelling, Optimization and Performance Analysis

The large diffusion of handheld devices is leading to an exponential growth of the mobile traffic demand which is already overloading the core network. To deal with such a problem, several works suggest to store content (files or videos) in small cells or user equipments. In this thesis, we push the idea of caching at the edge a step further, and we propose to use public or private transportation as mobile small cells and caches. In fact, vehicles are widespread in modern cities, and the majority of them could be readily equipped with network connectivity and storage. The adoption of such a mobile cloud, which does not suffer from energy constraints (compared to user equipments), reduces installation and maintenance costs (compared to small cells). In our work, a user can opportunistically download chunks of a requested content from nearby vehicles, and be redirected to the cellular network after a deadline (imposed by the operator) or when her playout buffer empties. The main goal of the work is to suggest to an operator how to optimally replicate content to minimize the load on the core network. The main contributions are: (i) *Modelling*. We model the above scenario considering heterogeneous content size, generic mobility and a number of other system parameters. (ii) *Optimization.* We formulate some optimization problems to calculate allocation policies under different models and constraints. (iii) *Performance analysis.* We build a MATLAB simulator to validate the theoretical findings through real trace-based simulations. We show that, even with low technology penetration, the proposed caching policies are able to offload more than 50 percent of the mobile traffic demand.

Résumé

Les Véhicules comme un Mobile Cloud: Modélisation, Optimisation et Analyse des Performances

La proliferation des appareils portables mene à une croissance du trafic mobile qui provoque une surcharge du coeur du réseau cellulaire. Pour faire face à un tel problème, plusieurs travaux conseillent de stocker les contenus (fichiers et vidéos) dans les small cells. Dans cette thèse, nous proposons d'utiliser les véhicules comme des small cells mobiles et de cacher les contenus à bord, motivés par le fait que la plupart des véhicules pourra facilement être équipée avec de la connectivité et du stockage. L'adoption d'un tel cloud mobile réduit les coûts d'installation et de maintenance et présente des contraintes énergétiques moins strictes que pour les small cells fixes. Dans notre modèle, un utilisateur demande des morceaux d'un contenu aux véhicules voisins et est redirigé vers le réseau cellulaire après une deadline ou lorsque son playout buffer est vide. L'objectif du travail est de suggérer à un opérateur comment répliquer de manière optimale les contenus afin de minimiser le trafic mobile dans le coeur du réseau. Les principales contributions sont: (i) *Modélisation*. Nous modélisons le scénario ci-dessus en tenant compte de la taille des contenus, de la mobilité et d'un certain nombre d'autres paramètres. (ii) *Optimisation*. Nous formulons des problèmes d'optimisation pour calculer les politiques d'allocation sous différents modèles et contraintes. (iii) *Analyse des performances*. Nous développons un simulateur MATLAB pour valider les résultats théoriques. Nous montrons que les politiques de mise en cache proposées dans cette thèse sont capables de reduire de plus que 50% la charge sur le coeur du réseau cellulaire.

Acknowledgements

I would like to express my gratitude to my supervisors *Akis* and *Chadi* for encouraging my Ph.D study, for the knowledge imparted and for allowing me to grow as a research scientist. Their guidance helped me in all the time of research and writing of this thesis. I must mention that they have been as two elder brothers for me rather than "simple" advisors, and that is the reason why I referred to them with their first names.

Besides my advisors, I would like to thank the rest of my thesis committee: Prof. *Emilio Leonardi*, Prof. *Iordanis Koutsopoulos*, Dr. *Georgios Paschos*, Prof. *Katia Obraczka*, Prof. *Petros Elia*, Prof. *Giovanni Neglia*, for their insightful comments and encouragement. My sincere thanks also goes to Prof. *Walid Dabbous* who provided me the opportunity to join the DIANA team. I would also like to thank all the colleagues at EURECOM and INRIA that collaborated and gave me fruitful insights for my research. Specifically, I thank my officemate and friend *Konstantinos* for the stimulating discussions and for the sleepless nights we were working together before deadlines which we sadly used to call "parties".

A special thanks goes to my family. Words cannot express how grateful I am to my mother *Miranda*, my father *Francesco* and my brother *Valentino* for supporting me throughout writing this thesis and in my life in general. Last but not least, I would like to thank my old friends *Andrea*, *Simone*, *Marco*, *Fabio*, *Mazza*, *Giorgio*, *Sara*, *Silvia*, *Shari*, *Gloria*, *Skina*, *Lorenzo*, who motivated me to strive towards my goal, and all of the new people I met during these three years. Each of them has been part of my life and has contributed to build what I am now. In sparse order, thanks to *Lea*, *Maja*, *Peppe*, *Pasquale*, *Riccardo*, *Fabio*, *Mario*, *Milica*, *Barbara*, *Vera*, *Roberta*, *Deborah*, *Anja*, *Dario*, *Alessia*, *Minja*, *Ilham*, *Nikos*, *Pavlos*, *Ruggero*, *Francesco*.

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.

"La sera, come tutte le sere, venne la sera. Non c'è niente da fare: quella é una cosa che non guarda in faccia nessuno. Succede e basta. Non importa che razza di giorno arriva a spegnere. Magari era stato un giorno eccezionale, ma non cambia nulla. Arriva e lo spegne. Amen. Così anche quella sera, come tutte le sere, venne la sera."

Contents

List of Figures

List of Tables

List of Abbreviations

BKP	**B**ounded **K**napsack **P**roblem
CAPEX	**CAP**ital **EX**penditure
CDF	**C**umulative **D**ensity **F**unction
CDN	**C**ontent **D**elivery **N**etwork
CoMP	**Co**ordinated **M**ulti-**P**oint
GP	**G**eometric **P**rogram
IID	**I**ndependent and **I**dentically **D**istributed
KKT	**K**arush-**K**uhn-**T**ucker
LFU	**L**east **F**requently **U**sed
LRU	**L**east **R**ecently **U**sed
MILP	**M**ixed-**I**nteger **L**inear **P**rogramming
MINLP	**M**ixed-**I**nteger **N**on**L**inear **P**rogramming
MNO	**M**obile **N**etwork **O**perator
NLP	**N**on**L**inear **P**rogramming
OPEX	**OP**perational **EX**penditure
PDF	**P**robability **D**ensity **F**unction
ProSe	**Pro**ximity **Se**rvice
qGC	**q**oE-aware content caching with **G**eneric **C**ontact model
QoE	**Q**uality **o**f **E**xperience
qSC	**q**oE-aware content caching for **S**ingle **C**ontact model
SC	content caching for **S**ingle **C**ontact model
SC+	enhanced content caching with **S**ingle **C**ontact model
SON	**S**elf **O**rganizing **N**etwork
TTL	**T**ime **T**o **L**ive
VC	**V**ideo **C**aching with low vehicle density
VC+	**V**ideo **C**aching with generic vehicle density

List of Symbols

	CONTROL VARIABLES	
$\mathbf{x}$	Number of replicas stored in vehicles	
X	Feasible region for $\mathbf{x}$	
$\mathbf{y}$	Set of deadlines	
Y	Feasible region for $\mathbf{y}$	$[s]$
	CONTENT	
k	Number of contents in the catalogue	
ϕ_i	Request rate for content i	
p_i	Probability to successfully download content i during a contact	
c	Buffer size per vehicle	
$\mathbf{s}$	Content size	$[MB]$
	MOBILITY	
T_{ij}	Inter-meeting time between $\mathcal{U}$ and $\mathcal{H}$ nodes	$[s]$
T_i	Inter-meeting time between $\mathcal{U}$ and any $\mathcal{H}$ nodes storing content i	$[s]$
λ	Mean inter-meeting rate between $\mathcal{U}$ and $\mathcal{H}$ nodes	$[s^{-1}]$
$\mathbf{E}[D]$	Mean contact duration	$[s]$
y_0	Maximum deadline	$[s]$
h	Number of vehicles	
M_i	Number of contacts with vehicles storing content i within deadline	
	CHUNK DOWNLOAD	
w_{ij}	Bytes downloaded per content	$[MB]$
μ	Mean of w_{ij}	$[MB]$
σ^2	Variance of w_{ij}	
W_i	Total bytes downloaded for content i from $\mathcal{H}$ nodes[1]	$[MB]$
f_{W_i}	Probability density function of W_i	
F_{W_i}	Cumulative density function of W_i	
	QUALITY OF EXPERIENCE PARAMETERS	
r_P	Mean viewing playout rate	$[Mbps]$
r_I	Mean download rate from the cellular infrastructure	$[Mbps]$
r_H	Mean download rate from the vehicular cloud	$[Mbps]$
$\mathbf{E}[Y_i]$	Expected bulk size	$[MB]$

[1] In Chapter 5, we consider the bytes downloaded from $\mathcal{I}$ nodes.

Ω	Mean slowdown	
y_{max}	Maximum deadline	$[s]$
ω_{max}	Upper bound on the mean slowdown	

QUEUEING PARAMETERS

$B_i^{(n)}$	Length of the nth busy period of the playout buffer	
$I_i^{(n)}$	Length of the nth idle period of the playout buffer	
e_i	Error introduced by the stationary assumption for content i	$[MB]$

SETS

$\mathcal{I}$	Infrastructure nodes
$\mathcal{H}$	Helper nodes
$\mathcal{U}$	End user nodes
$\mathcal{K}$	Content catalogue

A Mamma e Papà

Chapter 1

Introduction

1.1 Context

The last decade has seen an exponential growth in the mobile traffic demand. In particular, global mobile data traffic grew 63 percent in 2016 [Cisco, 2016-2021]. The ever changing mix and growth of wireless devices that are accessing mobile networks worldwide is one of the primary contributors to global mobile traffic growth [Analysis Mason, 2014]. By 2021, Cisco, 2016-2021 estimates 8,3 billion of handheld or personal mobile-ready devices and 3,3 billion of machine-to-machine connections (e.g., GPS systems in cars, asset tracking systems in shipping and manufacturing sectors). Figure 1.1 shows that mobile devices and connections are not only getting smarter in their computing capabilities but are also evolving from lower-generation network connectivity (2G) to higher-generation network connectivity (3G, 4G or LTE).

To sum up, *mobile data traffic is expected to grow to 49 exabytes*[1] *per month by 2021*, a sevenfold increase over 2016 (Figure 1.2). Such a demand (that will increase in the next years) is already overloading the cellular infrastructure.

In wired environments, content delivery networks (CDNs) have been successfully used for years (e.g., Akamai, Amazon CloudFront, Cloudflare) to decrease the load on the backbone [Borst, Gupta, and Walid, 2010]. A CDN consists of a large network of distributed local caches deployed in multiple data centers. The main goals are to improve network performance reducing the access content delay, and avoid congesting the central (back end) servers. Although CDNs are widely implemented in wired environments, wireless networks cannot take advantage of the same technology due to two main reasons: first, a single CDN data center corresponds to hundreds or thousands of small cells that require much higher deployment costs; second, wireless transmission is considerably more challenging than wired transmission, due to interference, but also offers opportunities due to its broadcast nature [Maddah-Ali and Niesen, 2015].

An attempt to reduce the load on the cellular infrastructure is given by the next generation of mobile technology 5G[2]. This standard should, at least, fulfill the following requirements compared to LTE: improved aggregated data rate (and peak rate),

[1] One exabyte corresponds to 10^{18} bytes.

[2] 5G is expected to be rolled out by 2020 [Alliance, 2015].

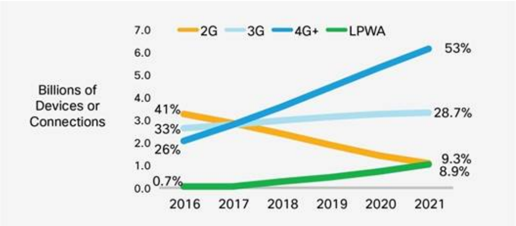

FIGURE 1.1: Global number of mobile devices and connections by 2G, 3G, 4G+ by 2021 [Cisco, 2016-2021].

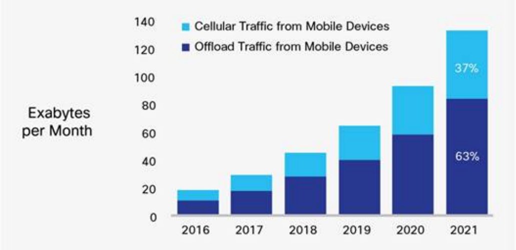

FIGURE 1.2: Cellular and offloaded traffic forecast by 2021 [Cisco, 2016-2021].

larger number of simultaneous connections for wireless sensors, enhanced spectral efficiency, reduced latency. Researchers are considering a number of new communication technologies to deal with such requirements [Andrews et al., 2014]:

- *Massive MIMO*. Multiple-antenna technology (i.e., MIMO) has already been incorporated into LTE and Wi-Fi. Advances in MIMO are required to use a very large number of service antennas to focus the transmission and reception of signal energy into smaller regions [Liu and Lau, 2014; Liu and Lau, 2015]. Hence, the main goal of massive MIMO is to increase spectral efficiency. Other benefits include the extensive use of inexpensive low-power components, reduced latency, and robustness to interference.

- *Millimetre-wave frequencies*. Millimetre wave spectrum is the band of spectrum between 30 GHz and 300 GHz. This higher frequency spectrum leads to a bandwidth increase that can be used for high-speed communications in order to accommodate the large traffic demand [Rappaport et al., 2013]. Today, these wave frequencies are mainly used indoor due to high propagation losses in outdoor environments.

- *Cognitive radio technology*. This allows different radio technologies to share the same spectrum efficiently. This is done by adaptively finding unused wireless channels and adapting the transmission scheme.

- *Small cell densification*. Cell sizes have been progressively shrinking in urban areas. Densification through small cells promises improved spectral efficiency at a smaller capital and operational expenditures (CAPEX and OPEX) [Andrews et al., 2014]. However, introducing a large number of small cells requires significant upgrades to the backhaul network which is predicted to become the new bottleneck [Forum, 2013; Sapountzis et al., 2016].

While 5G might offer good performance reducing the congestion, such upgrades are quite expensive. Due to the skewed nature of the Internet content popularity, several works have proposed *mobile data offloading* through caching in order to decrease the load on the cellular infrastructure without overloading the backhaul network [Han et al., 2012; Golrezaei et al., 2012; Wang et al., 2014a; Bastug, Bennis, and Debbah, 2014]. Caching content at the edge of the communication network (i.e., at a base station, small cell or mobile device) leads to a number of benefits for the performance of the network, namely reducing latency and saving bandwidth. Nevertheless, some have speculated that Wi-Fi offload will become less relevant for 4G networks because of the faster speeds and the more abundant bandwidth. However, 4G networks have attracted high-usage devices such as advanced smartphones and tablets, and now their plans are subject to data caps similar to 3G plans. For these reasons, Wi-Fi offloading is higher on 4G than on lower-speed networks, and the trend is expected to be similar for next generation networks [Cisco, 2016-2021]. What is more, while caching popular content deep inside the operator's core network promises good hit rates [Wang et al., 2014a; Erman et al., 2011; Woo et al., 2013], recent studies [Han, Liu, and Lau, 2016] show that caching at the base station is worthwhile due to its spectrum efficiency gain.

1.2 Problem Statement

Mobile edge caching can reduce the load on the cellular infrastructure, but also brings some main drawbacks. For instance, concerning content caching in small cells there are two main issues: (i) extensive small cell coverage is necessary to ensure enough traffic is offloaded from the macrocells [Robson, 2012], but initial experience with small cells suggests a bigger CAPEX/OPEX investment per site than initially predicted [Alliance, 2015]; (ii) clear benefits by edge caching are yet to be demonstrated, and initial studies based on real data are pessimistic [Paschos, Gitzenis, and Tassiulas, 2012]. This is due to smaller cache sizes and a much smaller overlap among user requests at a given cell, compared to an aggregation point deep inside the network. On the other hand, caching content at user equipments and using local device-to-device communications [Bao et al., 2013; Han et al., 2012] is a low cost solution. While more affordable for an operator, device-based caching faces significant technology adoption concerns as user equipments have limited storage capacity and strict battery constraints.

As we have already explained, the current cellular network is overloaded by the large mobile traffic demand. Hence, the main goal of the thesis is to *suggest to an operator how to reduce the load on the cellular infrastructure through mobile edge caching*. The proposed solution must satisfy three fundamental requirements:

- *Limiting CAPEX/OPEX*. Mobile network operators (MNOs) would incur significantly higher CAPEX and OPEX for 5G as they need to deploy thousands of small cells linked up with high bandwidth connections. Thus, limiting equipment costs, delaying hardware investments and reducing power consumption are necessary to adopt new generation technologies.

- *Storage capacity*. Due to the vastness of the Internet catalogue, large storage capacity is needed to ensure a large number of cache hits, and, thus, a higher percentage of traffic offloaded.

- *User Quality of Experience (QoE)*. In the context of telecommunications, the QoE is defined as the degree of delight or annoyance of the user of an application or service [Brunnström et al., 2013]. A new network technology must necessarily take into account the impact that such a solution would have on the users.

While there are a number of technical challenges to consider (concerning implementation and protocols), this work is mainly focused on analytical *study*, *modelling*, and *optimization*. Architectural details will be discussed in Appendix A.

1.3 Contributions of the Thesis

In order to deal with the aforementioned requirements, we propose to use private (e.g., cars) and/or public (e.g., taxis, buses) transportation acting as mobile small cells where to store popular content. The whole set of vehicles participating in the offloading mechanism forms a *vehicular cloud*. In the proposed infrastructure, an MNO pushes popular content in the vehicular cloud to offload part of the traffic. Differently from small cell deployments, the vehicular cloud brings three fundamental advantages:

- *Vehicle mobility virtually extends the size of accessible local storage.* With fixed caches, the amount of data offloaded depends (almost exclusively) on the femto or picocells coverage, since most of users exhibit a nomadic behavior, staying in the same location for long periods. On the other hand, a user will encounter several vehicles during the content download, especially in a dense urban environment, thus virtually extending the size of the accessible local storage. Simulations confirm that the vehicular cloud can reduce the load on the infrastructure, compared to the case of static small cell caches.

- *Vehicular cloud reduces CAPEX/OPEX compared to caching in small cells.* The current cellular infrastructure can be easily turned into a working vehicular cloud. The fundamental hardware components needed are the support for vehicle-to-user equipment and vehicle-to-infrastructure communications (e.g., 802.11p, LTE-A) and storage capacity, usually available at low cost. Sensors (e.g., GPS) can be considered as a plus to gather additional information. Basic computational capacities are required, such as an authentication system, and a connection manager for heterogeneous networks.

- *Vehicular cloud opens the market to new MNOs.* The simplicity to turn modern cities in a vehicular cloud may encourage new mobile virtual network operators to enter into the market without the need of large investments. This can be useful, e.g., in developing countries, where the data demand increases at the same rate as in the developed countries: thanks to the proposed infrastructure, new operators can boost the current cellular infrastructure at low cost, and, thus, improve the user QoE.

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 [Green, 2014]. For instance, BMW, that has already been embedding SIM cards for mobile connectivity in all its new cars [*Smartphone on*

Wheels], 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 [*BMW Vehicular CrowdCell*]. This project is in collaboration with Vodafone. 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. 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 (e.g., traffic, waste collection).

We consider the vehicular cloud as an additional feature to boost cellular data plans. When a user browses the Internet, the MNO might decide to redirect the requests to the vehicles for popular content (as it happens in the CDN context), if the user has subscribed for the vehicular cloud additional feature. While the architectural details of such a hybrid system are beyond the scope of the thesis and will be discussed in Appendix A, we sketch here two possible implementations in near future wireless systems:

- *Device-to-device connection*. In the simplest case, vehicles could act as end users (i.e., user equipments) in an LTE system. In other words, the "backhaul" link of our hybrid system is just a regular downlink between an eNodeB (the standard LTE base station) and a vehicle, over which content is pushed during off-peaks[3]. The "fronthaul" link could then be operated as a device-to-device link. Bychkovsky et al., 2006 have confirmed the feasibility of opportunistic connections between vehicles and user equipments. IEEE 802.11p, which has been developed for the specific context of vehicular networks, is the de facto standard offering simplicity (uncoordinated access mechanism, no authentication) and low delay (few hundreds milliseconds in crowded areas).
- *Full LTE integration*. As an alternative, a higher integration with cellular infrastructure could also be envisioned, where the vehicles are operated as (mobile) LTE relays [Sesia, Toufik, and Baker, 2009] with local caches, and end users devices as regular user equipments that can communicate with both macrocells and relays.

For simplicity, in our discussion we implicitly assume the former type of setup in order to avoid to deal with cross-interference issues between backhaul and fronthaul

[3] Note that, unlike static small cells, content cannot usually be pushed at night to our mobile helpers, except for taxis or night buses. Nevertheless, we believe that, within the window of 24 hours, there are enough traffic troughs to be able to update the helper caches from day to day without congesting the network.

links. However, futuristic tightly integrated architectures like the latter could offer the MNO better option to coordinate interference.

In this work, we exploit such a vehicular cloud to store popular content in order to *maximize the offloaded mobile demand*. We build a model where a user can opportunistically download content or part of it (i.e., chunks) from vehicles in her communication range. Mobile caches introduce interesting content mixing properties leading to larger hit rates than fixed small cells as we will explain in detail in the rest of the thesis. What is more, vehicles intrinsically satisfy two out of three requirements listed in the previous section as they are inexpensive and can be easily equipped with large storage capacity. To deal with the user experience, we provide different allocation policies depending on an increasing level of perceived user QoE along with a finer-grained granularity of the content download model. We list here the main contributions of the work:

- *Modelling.* We model the aforementioned vehicular cloud and the related communication with end users. Such an interaction is challenging due to the intrinsic mobility of the nodes, and the consequent intermittent availability of the download source. We do not make additional assumptions on the vehicle mobility patterns in order to consider any inter-contact time model. We use two models to deal with the content download: first, we introduce coarse-grained granularity where a content is entirely downloaded (with some probability) during a single contact; while this can be considered reasonable for small content, large content requires finer-grained granularity, and we introduce an enhanced model to consider downloads at chunk level.

- *Optimization.* According to different content download models and content types, we analytically formulate optimization problems to calculate the number of content items to allocate in the vehicular cloud so as to maximize the amount of data offloaded. We show the complexity of such problems, and we propose heuristics, continuous relaxations or approximations of the objective function in order to solve them efficiently.

- *Performance analysis.* We build a MATLAB tool to perform simulations based on real traces for vehicle mobility and content popularity in order to support our theoretical results. In the simulator, we consider a number of parameters such as user mobility, realistic cache sizes, rate adaptation and association setup mechanism during the content download. Furthermore, we provide some evidence that the vehicular cloud reduces CAPEX and OPEX comparing our architecture to close competitors.

1.4 Outline

The rest of the paper is structured as follows: in Chapter 2, we compare this thesis with some relevant related works; then we present our novel caching policies which are summarized in Table 1.1:

TABLE 1.1: Summary of the policies presented in the thesis (*SC+* is not listed since it coincides with *SC* with an improved content download model).

Policy	Vehicle mobility	Download model	Content type	User QoE
SC	Any	Content-level	Small content	Fixed deadlines
qSC	Any	Content-level	Small content	Variable deadlines
qGC	Any	Chunk-level	Any	Variable deadlines
VC	Sparse vehicles	Chunk-level	Video streaming	No deadlines
VC+	Any	Chunk-level	Video streaming	No deadlines

Chapter 3 - Content Caching through a Vehicular Cloud

We perform a preliminary study of the vehicular cloud based on two main ideas: (i) vehicles as mobile caches are more widespread and require lower costs compared to small cells; (ii) combining the mobility of vehicles with *delayed content access*, it is possible to increase the number of cache hits, and consequently to reduce the load on the infrastructure. Differently from fixed small cells, when caches are on vehicles, 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. Thus, in our system maximum delays are guaranteed to be kept to a few minutes. Beyond this deadline, the content is fetched from the infrastructure. We propose an analytical framework to compute the optimal number of content replicas that one should cache to minimize the infrastructure load. We assume that a content can be entirely downloaded during a single contact, and we present two caching policies (*SC* and *SC+*). Numerical simulations suggest that the vehicular cloud considerably reduces the infrastructure load in urban settings assuming modest penetration rates and tolerable content access delays.

The work related to this chapter is published in:

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Storage on wheels: Offloading popular contents through a vehicular cloud". In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2016, pp. 1–9. DOI: 10.1109/WoWMoM.2016.7523506

Chapter 4 - Quality of Experience-Aware Content Caching

In most delayed offloading settings, the worst-case delay deadline guarantee offered to the user is usually fixed for all content requests. Conversely, we propose to the operator to set different deadlines for different contents. Tuning the waiting time per content ensures maximum offloading with little QoE degradation that we evaluate according to the experienced slowdown which relates the waiting delay with the "net" download time. We model analytically such a scenario, and we formulate an optimization problem to maximize the traffic offloaded while ensuring user experience guarantees. We propose two variable deadline policies with file download at content-level (*qSC*) or chunk-level (*qGC*). Finally, we perform realistic trace-based simulations, and we show that, even with low technology penetration rate, more than 60 percent of the total traffic can be offloaded which is around 20 percent larger compared to existing allocation policies.

The works related to this chapter are published in:

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud". In: MSWiM (2017), *under review*

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud". In: *IEEE Transactions on Mobile Computing* (2017), *under review*

Chapter 5 - Content Caching for Video Streaming

The vast majority of the traffic concerns videos, and new streaming services have been recently introduced in the market (e.g., Netflix, Amazon Prime). In this chapter we argue that mobile caches can be used for low cost video streaming *without the need to impose any delay on the user*. Users can prefetch video chunks into their playout buffers from encountered vehicle caches (at low cost), or stream from the cellular infrastructure (at higher cost) when their playout buffers empty, while watching the content. We model the buffer dynamics as a queueing system, and analyse the characteristics of its idle periods (during which access to the cellular infrastructure is required). Based on this model, we formulate the problem of optimal allocation of content in vehicles to minimize the total load on the cellular infrastructure. We solve such an optimization problem for low or generic vehicle densities (*VC* and *VC*+). We perform trace-based simulations to support our findings, showing that up to 50 percent of the original traffic could be offloaded from the main infrastructure.

The works related to this chapter are published in:

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Streaming Content from a Vehicular Cloud". In: *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. CHANTS '16. New York City, New York: ACM, 2016, pp. 39–44. ISBN: 978-1-4503-4256-8. DOI: 10.1145/2979683.2979684. URL: http://doi.acm.org/10.1145/2979683.2979684

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Low Cost Video Streaming through Mobile Edge Caching: Modeling and Optimization". In: *IEEE Transactions on Mobile Computing* (2017), *under review*

Finally, we conclude our thesis in Chapter 6 with a summary and future work.

Chapter 2

Literature Review

The exponential growth of the mobile traffic is pushing academic research and companies to study new solutions to decrease the load on the cellular network: specifically, on the one hand, the networking research community is more interested in the theoretical problems of, e.g., content allocation, modelling, mobility pattern analysis; on the other hand, companies and MNOs see such a problem as an opportunity to fill the market gap with new products. In this chapter, we present a list of the most relevant and recent work in mobile edge caching: specifically, in Section 2.1, we discuss extensively about caching in small cell base stations which has been proposed as one of most interesting solutions for current and near future cellular networks. Then, we present two more futuristic approaches, such as caching on mobile devices (in Section 2.2) and caching on vehicular networks (in Section 2.3). Finally, we conclude the chapter in Section 2.4 with a review of two interesting theoretical works on the performance of caching systems. However, the analysis of the literature is not limited to this chapter, and additional references can be found throughout the thesis.

2.1 Caching at Small Cell Base Stations

Small cells constitute a promising solution to deal with the mobile data growth that is overloading the cellular network. Local caching of popular content items at the small cell base stations has been proposed to decrease the costly transmissions from the macrocell base stations without requiring high capacity backhaul links for connecting the small cells with the core network. In this context, traditional solutions concern adding storage capacity to small cell base stations [Golrezaei, Dimakis, and Molisch, 2012; Ao and Psounis, 2015] and/or to WiFi access points [Zhang et al., 2015] with a potential introduction of delay tolerance [Balasubramanian, Mahajan, and Venkataramani, 2010; Lee et al., 2013; Mehmeti and Spyropoulos, 2014] to further increase the number of cache hits. Due to the increasing diffusion of multimedia content, a number of works is also dealing with video caching recently [Poularakis et al., 2014; Dandapat et al., 2013]

2.1.1 Femtocaching

The femtocaching idea is proposed as a solution to compensate for the weak backhaul capacity, by deploying coverage-limited nodes with high storage capacity called femtocaches. The basic idea of caching in small cells has first been presented in Golrezaei et al., 2012; Shanmugam et al., 2013. Their main challenge is to analyse the optimum way of assigning content to the small cell base stations (called *helpers*) to minimize the expected download time or the number of cache hits. Such a generic formulation has later been improved and extended by several researchers considering different assumptions and models. For instance, Zhang et al., 2015 propose to cache content in wireless access points based on popularity. Also, the authors propose to separately add a prefetch buffer to capture short-term content access patterns using aggregated network-level statistics. Guan et al., 2014 exploit the mobility patterns of users assuming that these patterns are known a priori, and propose a heuristic to cache content. Blasco and Gündüz, 2014 assume that the content popularity is unknown and propose to refresh caches at regular intervals to learn the popularity profile.

Beyond femtocaching, researchers are also studying other techniques to improve the bandwidth capacity. Coordinated Multi-Point (CoMP) introduces a new approach where multiple available communications resources are being utilized to correspondingly transmit a signal by utilizing multiple base stations to transmit a signal to a user. Ao and Psounis, 2015 combine CoMP technologies with caching in small cells. The goal is to maximize the throughput of the system (i.e., higher content delivery speed and backhaul cost reduction) considering limited cache size, network topology and content popularity. An important aspect of the framework is the joint optimization of the cache allocation in the application layer and the cooperative transmission techniques (maximum radio transmission or zero-forcing beamforming).

However, other work does not aim to reduce the backhaul cost or to maximize the number of cache hits. For instance, the goal in Baştuğ, Guénégo, and Debbah, 2013 is to maximize a QoE metric, defined as the percentage of *satisfied requests*, subject to storage and backhaul capacity constraint. A request is satisfied when the average delivery rate is superior to a bandwidth requirement for such a request. The problem is solved through a simple greedy algorithm. Ostovari, Wu, and Khreishah, 2016 study the problem of collaborative caching in cellular network. The objective is to minimize the aggregated caching and download cost. The model considers *unlimited* cache space. Baştuğ, Bennis, and Debbah, 2014 model and characterize outage probability and average delivery rate as a function of the signal-to-interference-ratio, base station intensity, file bitrate, storage size and content popularity, but the work does not suggest content allocation. Finally, a survey on current techniques related to caching in current (and future 5G) mobile networks is presented in Wang et al., 2014a along with a novel edge caching scheme based on the concept of content centric networking.

While such distributed caching schemes for small cells provide very interesting theoretical insights and algorithms, they face some key shortcomings. A large number of small cells is required for an extensive enough coverage by small cells, which comes at a high cost [Alliance, 2015]. E.g., in a macro-cell of a radius of a few kilometers, it is envisioned to place from three to five small cells, 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 a limitation as explained next. Another key difference is that base stations 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 delay access which "mixes" base stations and users.

2.1.2 Delayed Content Access

To alleviate the aforementioned problem of request 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 [Balasubramanian, Mahajan, and Venkataramani, 2010; Lee et al., 2013; Mehmeti and Spyropoulos, 2014] or until to reach peer nodes in a peer-to-peer infrastructure [Cai, Koprulu, and Shroff, 2013; Sermpezis and Spyropoulos, 2014]. For example, Balasubramanian, Mahajan, and Venkataramani, 2010 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 by WiFi access points, allowing a larger ratio of connections to be offloaded than the mere physical coverage of WiFi access points allows. Li et al., 2011 propose a framework similar to our basic model that will be described in Chapter 3. Specifically, this work defines a system offloading utility function subject to individual linear constraints for storage capacity. Whitbeck et al., 2012 examine strategies to determine *how many* copies of the content should be injected, *when*, and to *whom*. This is achieved through a control loop that collects user-sent acknowledgements to determine if new copies need to be re-injected in the network. It is a generic delay-tolerant framework that can be applied to several scenarios (e.g., software and system updates, floating data). In other works [Cai, Koprulu, and Shroff, 2013; Gao et al., 2016], different deadlines are assigned to different contents. 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 work (see Chapter 4).

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 perceivable by the operator. Instead, having the small cells and caches move, naturally happening when placed on vehicles, allows the operator to offload more traffic with minimum (or no) QoE impact.

2.1.3 Video Caching

A number of works have recently focused their interest in caching of video content since multimedia files have become popular. For instance, Poularakis et al., 2014 optimize the service cost and the delivery delay. In their work, pre-stored video files can be encoded with two different schemes in various qualities. Differently, Dandapat et al., 2013 study just in time video streaming. The authors propose to exploit WiFi access points in cities to build a large distributed video caching system for smooth streaming. Since a user is not able to download an entire video from the same hotspot, the authors promote replication of video chunks across access points while aiming at minimizing this replication (efficient content storage). Such a problem is solved numerically and no insights concerning the optimal allocation are given. Furthermore, Ahlehagh and Dey, 2014 improve video capacity and user experience of mobile networks. The authors propose video-aware backhaul and wireless channel scheduling techniques to maximize the number of concurrent video sessions by satisfying QoE requirements. They consider video QoE as consisting of initial buffering delay and number of stalls during the video session. Finally, in the context of QoE, Tasiopoulos, Psaras, and Pavlou, 2014 model the intermittent availability of WiFi access point as a function of undisrupted video playback.

In Chapter 5, we introduce a model to deal specifically with multimedia content. This model is based on queueing theory and handles a number of system parameters such as heterogeneous content size, vehicle mobility and limited content capacity. According to such a model that exploits the intrinsic delay tolerance of later chunks, we are able to formulate an optimization problem, and solve it efficiently. The main novelties are: (i) the framework has a wide applicability: its generic formulation can be easily adapted to similar offloading scenarios such as femtocaching or caching on mobile devices; (ii) we combine the ideas of video cache at the edge and vehicular networks to improve the percentage of traffic offloaded; (iii) we provide insights on the solution of the optimization problem for different types of vehicle densities.

2.2 Caching on Mobile Devices

Apart from small cells, researchers have also been proposing to use mobile devices to offload content through opportunistic communications. Bao et al., 2013 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 device-to-device communications, exploiting the observation that cellular networks are strained when many people located in a small area request for content (e.g., concerts, sport stadiums, train stations). In Han et al., 2012, mobile devices store content and propagate the information opportunistically. Specifically, content delivery is delayed, and the challenge is to find a set of users where to offload the information. In addition, Cai, Koprulu, and Shroff, 2013 also take into account the time-varying channel conditions and the users' random mobility. Wang et al., 2014b analyze and model a framework where a set of mobile users receive content based on their content spreading impacts and their mobility patterns. Users share content via opportunistic connectivity with each other. Finally, the basic femtocaching framework [Golrezaei et al., 2012] has been extended to include caching at user equipments and device-to-device collaboration [Golrezaei et al., 2013; Golrezaei et al., 2014]. An extensive comparison of content caching on wireless devices techniques using the device-to-device approach with other alternative approaches (e.g., unicasting, harmonic broadcasting, coded multicasting) is provided in Ji, Caire, and Molisch, 2016.

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 a large hard disk and a simple access point (even with MIMO capabilities) 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. Finally, it is important to note that, while we present our approach based on a vehicular cloud (due to the reasons extensively explained), the same model also applies to device-based offloading.

2.3 Vehicular Networks

Recent technology advances allow car manufacturers to build vehicles smarter and more sophisticated. New vehicles are able to communicate each other to exchange information about security and traffic, and provide infotainment systems to passengers. While large setup delays might be an obstacle, recent protocols (e.g., DSRC) have been considerably boosting vehicular networks over the last ten years. For this reason, MNOs see vehicles as (i) new potential clients or as (ii) nodes to boost the current cellular infrastructure:

- *Vehicles as clients*. Cellular operators see vehicles as potential devices to connect to their network, and dedicated data plans have been proposed. What is more, WiFi offloading for moving vehicles poses unique challenges due to high mobility, and researchers have been interested to model and analyze such an environment [Cheng et al., 2014]. For instance, Mahmood et al., 2016 introduce a probabilistic model to cache content at the edge nodes in order to serve content requests from vehicles (or moving users). The model is based on vehicle mobility patterns and trajectories.

- *Vehicles as small cell base stations*. As an alternative, vehicles can be used as small cell base stations where to store popular content. The guidelines for the creation of such a mobile cloud formed by vehicles acting as a mobile multihop network can be found in Mamun, Anam, and Alam, 2012. First work in this direction has appeared in the late 2000s [Zhang, Zhao, and Cao, 2009; Zhao and Cao, 2008]. Zhang, Zhao, and Cao, 2009 propose a peer-to-peer scheme to

> improve the performance of content sharing in intermittently connected vehicular networks. Zhao and Cao, 2008 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. The state of the art of vehicle-to-x communications is extensively reviewed in Lu et al., 2014; Zheng et al., 2015.

The hype around vehicular networks as part of the cellular infrastructure has been confirmed by the interest of car manufacturers (e.g., *BMW Vehicular CrowdCell*) and by the launch of new companies (e.g., *Veniam*).

2.4 Performance of Caching Systems

Performance of caching systems is one of the most widely investigated topic in computer science. Nevertheless, probably due to its important role in Internet content delivery, caching has recently received a renewed interest by the networking research community. While the literature on caching is boundless, in this section we choose to review a few of recent relevant works which tackle such a problem from a generic and theoretical point of view. For instance, Dehghan et al., 2016 propose a utility-driven caching where a utility function is associate to each content. They formulate an optimization problem to maximize the aggregate content utility subject to capacity constraints. Interestingly, existing caching policies (e.g., LRU, FIFO) can be modelled within this framework.

In another approach, Garetto, Leonardi, and Traverso, 2015 introduce a way to analyze various caching strategies (e.g., LRU, q-LRU) when content popularity varies over time. The work considers separately single and interconnected caches, and is based on the "Shot Noise Model" [Traverso et al., 2013] to capture the dynamics of content popularity. In our work, the content popularity is considered "stable" during a time window defined by the operator. While this assumption might seem approximate, content popularity indeed varies slowly for some categories of content (e.g., videos, software updates). We discuss content popularity more in detail in Appendix B. What is more, although our focus is on the initial "one-shot" optimal allocation, we also provide a dynamic heuristic to update caches according to varying content popularity (see Appendix C).

Chapter 3

Content Caching through a Vehicular Cloud

3.1 Introduction

In this chapter, 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. Delayed offloading to small cells (with and without local storage) has already been considered (e.g., via WiFi access points) [Balasubramanian, Mahajan, and Venkataramani, 2010; Cai, Koprulu, and Shroff, 2013; Mehmeti and Spyropoulos, 2014; Han et al., 2012]. 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 time to live (TTL) in the order of half to a couple of hours to demonstrate performance benefits [Balasubramanian, Mahajan, and Venkataramani, 2010; Lee et al., 2013; Mehmeti and Spyropoulos, 2014].

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 3.4). In our system maximum delays are *guaranteed*, and kept to a *few minutes*: beyond a deadline agreed between the MNO and the user, the content is fetched from the infrastructure. Such additional waiting delays could be easily amortized for large content transmissions (e.g., videos, software downloads), or be acceptable based on user subscription level (e.g., some users might be willing to pay cheaper plans and live with the *occasional* longer delays [Ha et al., 2012]) and context (e.g., roaming users might be more willing to wait for a low cost access).

While there are a number of additional architectural and incentive-related questions to consider, the main goal of the work is to study how to optimally allocate content in order to minimize the load on the cellular infrastructure. The contributions of this chapter can be summarized as follows:

- We study analytically the problem of optimal content allocation, and derive the optimal number of copies of each content to be allocated in the vehicular cloud, assuming a stable catalogue and average content popularity.
- We extend our analysis to more practical settings where connectivity can be lost while retrieving the content (due to, e.g., mobility or interferences).
- We use real traces of content popularity and mobility to study the feasibility of our system, and show that considerable offloading gains can be achieved even with modest technology penetration (less than one percent of vehicles participating in the cloud) and reasonable *maximum* delays (one to five minutes).

Summarizing, this chapter is structured as follows: in Section 3.2, we present the content access protocol with the main assumptions of the model; next, in Section 3.3, we formulate an offloading optimization problem, and we provide closed-form expressions for the optimal allocation; then, we validate our results through simulations in Section 3.4; finally, we conclude with a summary in Section 3.5.

3.2 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.

3.2.1 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 downloaded).

The basic protocol is made up of three phases (Figure 3.1):

- ($\mathcal{I} \to \mathcal{H}$). $\mathcal{I}$ nodes place content in $\mathcal{H}$ nodes according to the chosen allocation policy. These allocation policies are the main outcome of this paper (specific policies will be discussed later). We refer to this phase as *seeding*. The seeding 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

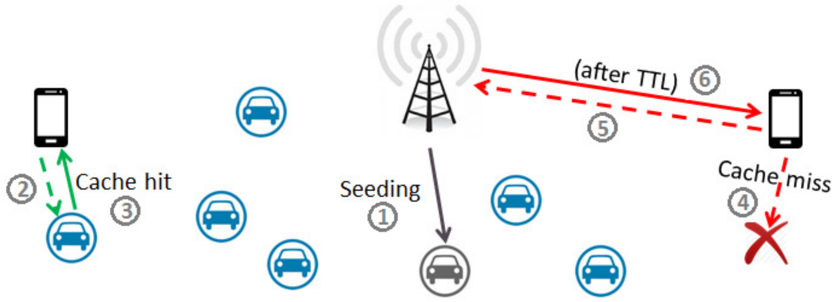

FIGURE 3.1: The basic communication protocol is defined by six steps: (1) the MNO pushes popular content in vehicles; (2) a user requests a content to nearby vehicle: (3) if the content is found, the user can immediately download it; (4) otherwise he waits for new vehicles. (5) When the deadline expires and the content has not been found, the user downloads it directly from the infrastructure.

periods, the seeding cost can be considered equal to 0. In our work, without loss of generality, we will focus on this scenario[1].

- ($\mathcal{H} \rightarrow \mathcal{U}$). An end user node can request content to the vehicles that are inside her communication range[2]. If content i is found, then the $\mathcal{U}$ node can immediately download it from the vehicle (with a certain probability, see following assumptions). Otherwise, she waits for new vehicles for a time equal to y_0. The related local access cost is assumed to be 0.
- ($\mathcal{I} \rightarrow \mathcal{U}$). In case of a content not successfully downloaded within y_0, the $\mathcal{U}$ node's request will be served 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 (i.e., the content size).

3.2.2 Main Assumptions

A.1 - Catalogue. Let $\mathcal{K}$ be the set of all possible contents that users might request (also defined as "catalogue"), where $|\mathcal{K}| = k$. 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 ϕ_i measured as the request rate within a seeding time window from all users and all cells. Similar to a number of works on edge caching [Golrezaei et al., 2013; Poularakis et al., 2014], 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

[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, and will be evaluated in the simulation section. A more detailed theoretical study is provided in Appendix C.1.

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

known, predicting the popularity of a content is more challenging. Nevertheless, 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 [Szabo and Huberman, 2010; Lee, Moon, and Salamatian, 2010]. Without loss of generality, we assume content is sorted by decreasing popularity as $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_k$.

A.2 - Mobility model. We assume that the inter-meeting times T_{ij} between a user requesting content $i \in \mathcal{K}$ and a vehicle $j \in \mathcal{H}$ are independent and identically distributed (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 $i \in \mathcal{K}$ 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 [Karagiannis, Boudec, and Vojnovic, 2010].

A.3 - Cache model. Let $x_{ij} \in \{0,1\},\ i \in \mathcal{K}, j \in \mathcal{H}$ 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 denote the number of $\mathcal{H}$ nodes storing content i:

$$x_i = \sum_{j \in \mathcal{H}} x_{ij}.$$

The vector $\mathbf{x}$ 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 - Content download. Opportunistic meetings between $\mathcal{U}$ and $\mathcal{H}$ are described by the well-known "protocol model" [Golrezaei et al., 2014; Gupta and Kumar, 2000] that uses a simplified description of the physical layer: specifically, two nodes can 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*. What is more, we assume that a download may fail during a contact. This can be due to several reasons, e.g., low signal-to-interference-plus-noise-ratio between the vehicle and the user, limited contact duration between the two, interference. In our framework, we assume that the download of the content will restart from the beginning when a $\mathcal{U}$ node loses the connection with a vehicle (e.g., TCP session expires) and meets a new one. Hence, we assume that each contact with a vehicle storing content i is successful with some mean probability $p_i \leq 1$. We define this contact download model as follows:

Definition 3.1 (Single Contact Model)**.** *Let a content be entirely downloaded during a contact with probability p_i. Assume also that the download restarts if content is not successfully downloaded. We refer to this scenario as "Single Contact Model".*

In the next chapter, we consider the possibility to resume the download for subsequent meetings. The notation used in the chapter is summarized in Table 3.1.

TABLE 3.1: Notation used in the chapter.

CONTROL VARIABLE	
x_i	Number of replicas stored for content i
X	Feasible region for $\mathbf{x}$
CONTENT	
k	Number of contents in the catalogue
ϕ_i	Request rate for content i
s_i	Size of content i
p_i	Probability to successfully download content i during a contact
c	Buffer size per vehicle
MOBILITY	
T_{ij}	Inter-meeting time between $\mathcal{U}$ and $\mathcal{H}$ nodes
T_i	Inter-meeting time between $\mathcal{U}$ and any $\mathcal{H}$ nodes storing content i
λ	Mean inter-meeting rate between $\mathcal{U}$ and $\mathcal{H}$ nodes
h	Number of vehicles
y_0	Maximum deadline
M_i	Number of contacts with vehicles storing content i within y_0
SETS	
$\mathcal{I}$	Infrastructure nodes
$\mathcal{H}$	Helper nodes
$\mathcal{U}$	End user nodes
$\mathcal{K}$	Content catalogue

3.3 Optimal Content Allocation with Single Contact Model

Based on the aforementioned content access protocol and on the previous assumptions, in Section 3.3.1 we formulate an optimization problem to reduce the load on the cellular infrastructure. Then, we propose specific policies to optimally cache popular content in the vehicular cloud when the failure download probability is equal to one (Section 3.3.2) or smaller (Section 3.3.3).

3.3.1 Offloading 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 contents 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.

Problem 1. *Consider the Single Contact Model. The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud*[3]*:*

$$\begin{aligned} \underset{\mathbf{x}\in X^k}{\textit{maximize}} \quad & \Phi(\mathbf{x}) = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - \sum_{j=0}^{+\infty} \mathbf{P}[M_i(y_0) = j] \cdot (1-p_i)^j \right), \qquad (1) \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h, \end{aligned}$$

where $X \triangleq \{a \in \mathbb{N} \mid 0 \leq a \leq h\}$ is the feasible region for the control variable $\mathbf{x}$, and $M_i(t)$ is a point process counting the contacts with vehicles storing content i within t.

The objective function counts the number of bytes offloaded through the vehicular cloud in a seeding time window for the entire catalogue. For each content, this is equivalent to the content size s_i times its request rate ϕ_i multiplied by the probability to *successfully* find it within y_0: this is equal to one minus the probability that j downloads all fail to find, for any $j > 0$, where j counts the number of contacts within y_0. The objective function is subject to the constraints:

- The number of replicas of content i cannot be negative:

$$x_i \geq 0, \quad \forall i \in \mathcal{K}.$$

[3] Note that maximize the number of bytes offloaded through the vehicular cloud is equivalent to minimize the bytes downloaded from the cellular infrastructure when the seeding cost is null.

- The number of replicas of content i cannot be higher than the number of vehicles participating in the cloud:
$$x_i \leq h, \quad \forall i \in \mathcal{K}.$$

- Each vehicle has a storage constraint and cannot store more than c contents. However, instead of considering h individual storage constraints (i.e., $\sum_{i=1}^{k} s_i \cdot x_{ij} \leq c$, $\forall j \in \mathcal{H}$), we only consider the global cache capacity of the vehicular cloud that corresponds to
$$\mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h$$
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 Section 3.3.2 (see randomized rounding).

Especially in urban environments, we note that the number of vehicles participating in the cloud is expected to be large, and the inter-meeting rate per vehicle is low. The following lemma describes the distribution of T_i under the above assumptions.

Lemma 3.2. *Assume the number of vehicles participating in the vehicular cloud to be large, and the mean inter-meeting rate with such vehicles small. Thus, T_i approaches an exponential distribution with rate $\lambda \cdot x_i$.*

Proof. Let $\Gamma_{ij}(t)$ be a point process with rate λ. Each point of this process is the time of a contact between a user and a vehicle $j \in \mathcal{H}$ storing content $i \in \mathcal{K}$. The inter-arrival time between two points is captured by the random variable T_{ij} (see Assumption A.2). To clarify, assume $T_{ij}^{(1)}$ to be a random variable that corresponds to the time of the first jump. Then, the event $\{\Gamma_{ij}(t) = 0\}$ is equivalent to the event $\{T_{ij}^{(1)} > t\}$, meaning that the first jump will occur after epoch t, i.e.,

$$\mathbf{P}[T_{ij}^{(1)} > t] = \mathbf{P}[\Gamma_{ij}(t) = 0].$$

Since content i has x_i replicas, there are x_i identical processes $\Gamma_{ij}(t)$. We equivalently redefine $\Gamma_{ij}(t)$ as follows: $\{\Gamma_{ij}(t), t > 0, j \in \mathcal{H} \mid x_{ij} = 1\}$ are x_i identical and independent renewal processes with *holding times* T_{ij} corresponding to the inter-arrival times between users and vehicles storing content i. Let further $\{\Gamma_i(t), t > 0\}$ be the superposition of these processes. According to the Palm-Khintchine theorem [Karlin and Taylor, 2012], $\{\Gamma_i(t), t > 0\}$ approaches a Poisson process with rate $\lambda \cdot x_i$ if x_i large and λ small. Note that T_i corresponds to the inter-arrival times of the process $\{\Gamma_i(t), t > 0\}$ (assumption A.2). Thus, T_i approaches an exponential distribution with mean rate $\lambda \cdot x_i$. □

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 [Conan, Leguay, and Friedman, 2007; Karagiannis, Le Boudec, and Vojnović, 2007]. In an urban environment, the above lemma approximates T_i accurately as we will verify through real trace-based simulations.

3.3.2 Content Caching with Single Contact Model (SC)

In this section, we assume that 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 (for instance, in the case of short videos, news or even advertisements) or (ii) the contact duration is large due to future envisioned improvements in vehicle-to-device communications. Therefore, we initially set p_i equal to one. This scenario will be considered as a baseline scenario.

Lemma 3.3. *Consider the Single Contact Model when $p_i = 1$ for any $i \in \mathcal{K}$. The objective function of Problem (1) becomes*

$$\Phi_{sc}(\mathbf{x}) = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - e^{-\lambda \cdot x_i \cdot y_0}). \tag{2}$$

Proof. When $p_i = 1$, Eq. (1) becomes

$$\Phi(\mathbf{x}) \equiv \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - \mathbf{P}[M_i(y_0) = 0]) \,.$$

$\mathbf{P}[M_i(y_0) = 0]$ is the probability not to have a meeting with a vehicles storing content i within y_0, and it is clearly equivalent to $\mathbf{P}[T_i > y_0]$. What is more, T_i is exponentially distributed with rate $\lambda \cdot x_i$ (see Lemma 3.2). Under these considerations, the objective function of Problem (1) becomes

$$\begin{aligned}\Phi_{sc}(\mathbf{x}) &= \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - \mathbf{P}[T_i > y_0]) \\ &= \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - e^{-\lambda \cdot x_i \cdot y_0}\right),\end{aligned}$$

which is the objective function of the lemma. □

Hence, Problem (1) can be rewritten as follows:

Problem 2. *Consider the Single Contact Model when $p_i = 1$ for any $i \in \mathcal{K}$. The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud:*

$$\begin{aligned}&\underset{\mathbf{x} \in X^k}{\textit{maximize}} && \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - e^{-\lambda \cdot x_i \cdot y_0}),\\ &\textit{subject to} && \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h.\end{aligned}$$

Proposition 3.4. *Problem (2) is an NP-hard combinatorial problem.*

Proof. The problem is a bounded knapsack problem (BKP) with a nonlinear objective function. Such a problem is NP-hard since it is a generalization of the knapsack problem which is known to be NP-hard [Martello and Toth, 1990]. □

Similarly to a number of works, we consider the *continuous relaxation* of the problem to obtain a closed-form real-valued solution. This 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 original integer problem. The following theorem holds:

Theorem 3.5. *The solution of 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 $\mathbf{x}^* \triangleq \arg\max_{\mathbf{x} \in X^k} \Phi_{sc}(\mathbf{x})$, $L \triangleq \frac{\rho}{\lambda \cdot y_0}$, $U \triangleq \frac{\rho \cdot e^{h \cdot \lambda \cdot y_0}}{\lambda \cdot y_0}$, *and* ρ *is an appropriate Lagrange multiplier.*

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

$$\begin{cases} l_i \cdot x_i = 0 \\ m_i \cdot (h - x_i) = 0 \\ \rho \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i\right) = 0 \end{cases}$$

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

$$\mathcal{L}(\mathbf{x}) = \sum_{i=1}^{k} \left[\phi_i \cdot s_i \cdot (1 - e^{-\lambda \cdot x_i \cdot y_0}) + l_i \cdot x_i + m_i \cdot (h - x_i)\right] + \rho \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i\right).$$

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.

$$\frac{d\mathcal{L}(\mathbf{x})}{dx_i} = \lambda \cdot y_0 \cdot \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i \cdot y_0} + l_i - m_i - \rho \cdot s_i = 0.$$

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

$$x_i = \frac{1}{\lambda \cdot y_0} \cdot \ln\left(\frac{\lambda \cdot y_0 \cdot s_i \cdot \phi_i}{s_i \cdot \rho - l_i + m_i}\right).$$

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) $x_i > 0,\ l_i = 0$; (ii) $x_i = 0,\ l_i > 0$. 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 $x_i > 0$:

$$\frac{1}{\lambda \cdot y_0} \ln\left(\frac{\lambda \cdot y_0 \cdot \phi_i}{\rho}\right) > 0 \Leftrightarrow \phi_i > \frac{\rho}{\lambda \cdot y_0} \triangleq L.$$

- *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) $x_i = h,\ m_i > 0$. Again, the threshold between case (i) and (ii) depends on the content popularity: specifically, a content will get less than h copies when its popularity is lower than U which can be easily computed when $x_i < h$:

$$\frac{1}{\lambda \cdot y_0} \ln\left(\frac{\lambda \cdot y_0 \cdot \phi_i}{\rho}\right) < h \Leftrightarrow \phi_i < \frac{\rho \cdot e^{h \cdot \lambda \cdot y_0}}{\lambda \cdot y_0} \triangleq U.$$

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

□

Corollary 3.6. *The* relative *optimal content allocation for Problem (2) is independent of the content size.*

When deadlines are fixed, the number of copies to allocate only depends on the content popularity. However, the *absolute* allocation[4] is still dependent on the content size: intuitively, if content has larger size, for a given popularity the optimal number of copies to allocate is lower due to the capacity constraint. To be more precise, the value of the Lagrange multiplier ρ permits to satisfy the capacity constraint (see KKT conditions in the proof of Theorem 3.5) by scaling the total allocation down. However, since ρ depends on the popularity of the entire catalogue, it is in general difficult to determine a closed-form expression for it. The following proposition helps to formulate a simple algorithm to find numerically its value.

Proposition 3.7. *The value of the objective function of Problem (2) monotonically decreases as ρ increases*[5].

Proof. Let $\mathbf{x}^0$ be the allocation computed when $\rho = \rho^*$ where $\rho^* > 0$ is an arbitrary positive number. Similarly, let $\mathbf{x}^1$ be the allocation computed when $\rho = \rho^* + \Delta\rho$. We

[4] With the term *absolute* allocation we refer to the actual number of replicas to cache in the vehicular cloud.

[5] Note that the optimal content allocation $\mathbf{x}^*$ is function of ρ.

want to prove that

$$\Phi_{sc}(\mathbf{x}^0) \;\geq\; \Phi_{sc}(\mathbf{x}^1),$$

for any $\Delta\rho > 0$. From Lemma 3.3, we have

$$\Phi_{sc}(\mathbf{x}^0) = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - e^{-\lambda \cdot x_i^0 \cdot y_0}\right),$$

$$\Phi_{sc}(\mathbf{x}^1) = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - e^{-\lambda \cdot x_i^1 \cdot y_0}\right).$$

Then,

$$\sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - e^{-\lambda \cdot x_i^0 \cdot y_0}\right) \geq \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - e^{-\lambda \cdot x_i^1 \cdot y_0}\right)$$

$$\sum_{i=1}^{k} \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i^0 \cdot y_0} \leq \sum_{i=1}^{k} \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i^1 \cdot y_0}$$

$$\sum_{i=1}^{k} \left(e^{-\lambda \cdot x_i^0 \cdot y_0} - e^{-\lambda \cdot x_i^1 \cdot y_0}\right) \leq 0.$$

It is easy to see that, for a given content i, $x_{1i} \geq x_{2i}$, for any $\Delta\rho > 0$. Thus,

$$e^{-\lambda \cdot x_i^0 \cdot y_0} - e^{-\lambda \cdot x_i^1 \cdot y_0} \leq 0, \;\forall i \in \mathcal{K},$$

which proves the proposition. □

Corollary 3.8. *Problem (2) is maximized for the minimum value of ρ that satisfies the capacity constraint.*

The above corollary can be directly derived from Proposition 3.7. Thus, we can write the following minimum search algorithm (Algorithm 1) to compute the optimal allocation:

1. Set ρ to an arbitrary low value ρ_0.
2. Calculate the optimal allocation $\mathbf{x}$ according to Theorem 3.5.
3. If the total allocation $\mathrm{s}^t \cdot \mathbf{x}$ is smaller than the total buffer capacity available $c \cdot h$, then go to 4. Otherwise, increase ρ and repeat 2.
4. Introduce the auxiliary variables a and b, and set ρ to $(a+b)/2$.
5. Calculate the optimal allocation $\mathbf{x}$ according to Theorem 3.5.
6. If the convergence criterion is satisfied (see below), then stop and return $\mathbf{x}$. Otherwise updates a, b and ρ, and go to 5.

Algorithm 1 Caching algorithm for Single Contact Model

```
Ensure: x
function MAIN
    $\rho \leftarrow \rho_0$                                  ▷ Set an initial value of $\rho$
    x← Compute_allocation($\rho$)                             ▷ See Theorem 3.5
    while $\mathbf{s}^t \cdot \mathbf{x} > \mathtt{c} \cdot \mathtt{h}$ do
        $\rho \leftarrow \rho \cdot 2$                        ▷ Increase $\rho$ arbitrarily
        x← Compute_allocation($\rho$)
    $\epsilon \leftarrow$0,001                                ▷ Buffer capacity unused less than 0,1%
    $\mathtt{a} \leftarrow \rho/2$
    $\mathtt{b} \leftarrow \rho$
    while $\mathbf{s}^t \cdot \mathbf{x} < \mathtt{c} \cdot \mathtt{h} \cdot (1-\epsilon)$ OR $\mathbf{s}^t \cdot \mathbf{x} > \mathtt{c} \cdot \mathtt{h}$ do   ▷ Convergence criterion
        $\rho \leftarrow (\mathtt{a}+\mathtt{b})/2$           ▷ Binary search
        x← Compute_allocation($\rho$)
        if $\mathbf{s}^t \cdot \mathbf{x} > \mathtt{c} \cdot \mathtt{h}$ then
            $\mathtt{a} \leftarrow \rho$
        else
            $\mathtt{b} \leftarrow \rho$
    return x
```

Let $\epsilon > 0$ be the percentage of storage capacity that can be left unused. Algorithm 1 stops when the convergence criterion is met, i.e., when

$$(1-\epsilon) \cdot c \cdot h \leq \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h.$$

Finally, we use *randomized rounding* [Raghavan and Tompson, 1987] 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). We refer to this policy as *Content Caching with Single Contact Model* (SC).

Content Popularity from a Known Distribution

In some cases, if the content popularity ϕ_i follows a known distribution, then it is possible to calculate ρ explicitly without the need for Algorithm 1. We provide an example when the content popularity follows a Pareto distribution.

Proposition 3.9. *Let $f_\phi(t)$ be a probability density function (PDF) that follows a Pareto distribution with shape α and scale x_m, i.e.,*

$$f_\phi(t) = \begin{cases} \frac{\alpha \cdot x_m^\alpha}{t^{\alpha+1}}, & \textit{if } t \geq x_m, \\ 0, & \textit{otherwise.} \end{cases}$$

Let further $f_\phi(t)$ be the PDF of the content popularity. Hence, Theorem 3.5 optimally solves Problem (2) when

$$\rho = \lambda \cdot y_0 \cdot x_m \cdot \sqrt[\alpha]{\frac{k \cdot \mathbf{E}[s_i]}{c \cdot h} \cdot \left[\frac{1}{\alpha \cdot \lambda \cdot y_0} \cdot [1 - (\alpha \cdot h \cdot \lambda \cdot y_0 + 1) \cdot e^{-\alpha \cdot h \cdot \lambda \cdot y_0}] + e^{-\alpha \cdot h \cdot \lambda \cdot y_0}\right]}.$$

Proof. Theorem 3.5 splits content in three regimes according to their popularity. Thus, we define the following disjoint sets:

$$\begin{aligned} \mathcal{A} &\triangleq \{a \in \mathcal{K} \mid \phi_a < L\}, \\ \mathcal{B} &\triangleq \{b \in \mathcal{K} \mid L \leq \phi_b \leq U\}, \\ \mathcal{C} &\triangleq \{c \in \mathcal{K} \mid \phi_c > U\}. \end{aligned}$$

where $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ correspond respectively to the subset of contents having low, medium and high popularity. According to Corollary 3.8, we want to calculate

$$\rho^* = \min\{\rho \in \mathbb{R}^+ \mid \mathbf{s}^t \cdot \mathbf{x}^*(\rho) \leq c \cdot h\},$$

which is minimum when the constraint has the equality (see Proposition 3.7). Thus, according to Theorem 3.5 and to the sets previously defined, we write the total allocation as a function of ρ:

$$\sum_{i \in \mathcal{A}} s_i \cdot 0 + \sum_{i \in \mathcal{B}} \frac{s_i}{\lambda \cdot y_0} \cdot \ln\left(\frac{\lambda \cdot y_0 \cdot \phi_i}{\rho}\right) + \sum_{i \in \mathcal{C}} s_i \cdot h = c \cdot h,$$

Without loss of generality, the corresponding continuous form is

$$k \cdot \int_L^U \frac{s_i}{\lambda \cdot y_0} \cdot \ln\left(\frac{\lambda \cdot y_0 \cdot t}{\rho}\right) \cdot f_\phi(t)\, dt + k \cdot \int_U^{+\infty} s_i \cdot h \cdot f_\phi(t)\, dt = c \cdot h.$$

Since the content catalogue is large and the content size is independent of its popularity, we can rewrite the above equation replacing s_i with its expected value $\mathbf{E}[s_i]$:

$$\frac{1}{\lambda \cdot y_0} \cdot \int_L^U \ln\left(\frac{\lambda \cdot y_0 \cdot t}{\rho}\right) \cdot f_\phi(t)\, dt + h \cdot \int_U^{+\infty} f_\phi(t)\, dt = \frac{c \cdot h}{k \cdot \mathbf{E}[s_i]}.$$

When $f_\phi(t)$ follows a Pareto distribution, the first integral can be easily solved by parts and is equal to

$$\left(\frac{\lambda \cdot y_0 \cdot x_m}{\rho}\right)^\alpha \cdot \frac{1}{\alpha \cdot \lambda \cdot y_0} \cdot \left[1 - (\alpha \cdot h \cdot \lambda \cdot y_0 + 1) \cdot e^{-\alpha \cdot h \cdot \lambda \cdot y_0}\right],$$

and the second integral is the complementary CDF of $f_\phi(t)$ calculated in U, i.e.,

$$\left(\frac{x_m \cdot \lambda \cdot y_0}{\rho \cdot e^{h \cdot \lambda \cdot y_0}}\right)^\alpha.$$

Then, solving for ρ gives the result of the proposition. □

3.3.3 Enhanced Content Caching with Single Contact Model (SC+)

When a $\mathcal{U}$ node is inside the communication range of an $\mathcal{H}$ node, the connectivity might be lost due to many reasons. Specifically, a success or a failure in downloading a content depends on the following key factors:

- *Contact duration*. This is the amount of time during which $\mathcal{H}$ and $\mathcal{U}$ nodes can exchange data during a contact. Due to vehicles mobility, an $\mathcal{H}$ node could leave the communication range before the content is entirely downloaded.
- *Throughput*. The download rate depends both on distance between the nodes and on interferences and variability in the urban radio environment: WiFi protocols are defined to use dynamic rate scaling, and the throughput will automatically decrease as the signal strength decreases, i.e., as the distance between the nodes increases.
- *Content size*. The probability to successfully download a content decreases as its size increases.

In order to deal with such a failure probability in the downloads, we will assume to have knowledge of an average value p_i of the probability to successfully download content i during a contact. Similarly to the previous section, we infer the objective function of the problem.

Lemma 3.10. *Consider the Single Contact Model when $p_i < 1$. The objective function of Problem (1) becomes*

$$\Phi_{sc+}(\mathbf{x}) = \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - e^{-p_i \cdot \lambda \cdot x_i \cdot y_0}).$$

Proof. In the Single Contact Model, a content is downloaded when at least one of the contacts is successful. When $p_i = 1$ for any $i \in \mathcal{K}$, $\{\Gamma_i(t), t > 0\}$ approximates a Poisson process with rate $\lambda \cdot x_i$ (see Lemma 3.2). Now, we assume that each arrival, independently of the the others, is one of two types: case 1 (*success*) with probability p_i, case 2 (*fail*) with probability $1 - p_i$. The new random process is referred to as *thinning* the original Poisson process. Hence, the two cases form separate Poisson processes with rates $p_i \cdot \lambda \cdot x_i$ and $(1 - p_i) \cdot \lambda \cdot x_i$ respectively, and are independent. □

Hence, Problem (1) can be rewritten as follows:

Problem 3. *Consider the Single Contact Model when $p_i < 1$. The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud:*

$$\begin{aligned} \mathop{\textit{maximize}}_{\mathbf{x} \in X^k} \quad & \sum_{i=1}^{k} \phi_i \cdot s_i \cdot (1 - e^{-p_i \cdot \lambda \cdot x_i \cdot y_0}), \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \le c \cdot h. \end{aligned}$$

Similarly to Problem (2), this problem is NP-hard, and we consider a continuous relaxation.

Theorem 3.11. *The solution of Problem (3) is given by*

$$x_i^* = \begin{cases} 0, & \text{if } \phi_i < L \\ \frac{1}{p_i \cdot \lambda \cdot y_0} \cdot \ln\left(\frac{p_i \cdot \lambda \cdot y_0 \cdot \phi_i}{\rho}\right), & \text{if } L \leq \phi_i \leq U \\ h, & \text{if } \phi_i > U \end{cases}$$

where $L \triangleq \frac{\rho}{p_i \cdot \lambda \cdot y_0}$, $U \triangleq \frac{\rho \cdot e^{h \cdot p_i \cdot \lambda \cdot y_0}}{p_i \cdot \lambda \cdot y_0}$.

Proof. The proof proceeds as in the proof of Theorem 3.5 when λ is replaced by $p_i \cdot \lambda$. □

We refer to this policy as *Enhanced Content Caching with Single Contact Model* (SC+).

3.4 Performance Analysis

We build a tool in MATLAB where we simulate the load on the cellular infrastructure in order to validate our model. We consider the proposed allocation policies *SC* and *SC+*, and we study the impact of different parameters in the proposed cache system.

3.4.1 Simulation Setup

We consider a square area 5000 m x 5000 m in the center of San Francisco. We use the Cabspotting[6] trace to compute the average inter-meeting rate between users and vehicles: we randomly place $\mathcal{U}$ nodes and, considering a communication range of 200 m, we calculate the meeting rate with each $\mathcal{H}$ node. We find $\lambda = 4$ contacts/day. According to the density of the city and to the number of vehicles per capita, we estimate to 100.000 the number of vehicles in the area considered. However, in order to be realistic about initial technology penetration, we assume that only one percent of these vehicles is participating in the cloud.

Furthermore, in our analysis we consider the content popularity of YouTube videos since a large percentage of mobile data traffic is represented by video files. We download a database that collects statistics for 100.000 YouTube videos [Zeni, Miorandi, and De Pellegrini, 2013]. The database includes static (e.g., title, description, author, duration, related videos) and dynamic information (e.g., daily and cumulative views, shares, comments). In our simulations, we only take into account the number of views related to one week. However, these values are equal to the total number of views per day in the world, then we scale them linearly taking into account the number YouTube users and the population of San Francisco. We have also created

[6] GPS coordinates from more than 500 taxis in San Francisco over approximately three weeks [Piorkowski, Sarafijanovic-Djukic, and Grossglauser, 2009].

synthetic traces based on the work in Crane and Sornette, 2008. Simulations based on these synthetic traces confirm the observations made using the real traces. We therefore focus on the former.

Finally, we assume that each car can store 100 contents ($0,1$ percent of the catalogue), and we set y_0 to three minutes. Unless otherwise stated, we will use these parameters summarized in Table 3.2.

DESCRIPTION	PARAM	VALUE
Number of vehicles	h	1000 vehicles
Buffer size	c	0,2%$\cdot k$
Meeting rate	λ	4 contacts/day
Deadline	y_0	3 minutes
Number of contents	k	100.000 contents
Simulation area		5000 m×5000 m
Communication range		200 m

TABLE 3.2: Parameters used in the simulations.

We compare the following allocation policies:

- *SC+*. This policy allocates content on vehicles proportionally to the logarithm of the popularity when the mean probability to download a content during a meeting is equal to p_i. The policy is described in Section 3.3.3.
- *SC*. This policy allocates content on vehicles proportionally to the logarithm of the popularity when the mean probability to download a content during a meeting is equal to one. The policy is described in Section 3.3.2.
- *Square root*. This policy behaves similarly to *SC*, but it replaces the logarithm with the square root, after an appropriate normalization to satisfy the storage constraint.
- *Random*. This policy allocates content randomly.
- *No cache*. No content is stored in the vehicles. The probability of miss is equal to one, therefore the cost corresponds to the total demand: cost $= \mathrm{s}^t \cdot \phi$.

3.4.2 Numerical Results

We perform numerical simulations comparing the effects of buffer size, deadlines and other parameters on the final gain comparing different policies.

Figure 3.2 depicts the cost in terms of percentage of bytes downloaded from the cellular infrastructure assuming $p_i = 1$ for any $i \in \mathcal{K}$. *SC* reduces the total cost by around 65 percent, more than any other policy. What is more, it improves twice the performance compared to the *square root* policy which is known to achieve optimal results in conventional peer-to-peer networks [Cohen and Shenker, 2002]. The bar chart also shows the seeding cost which is computed as the number of bytes pushed by the MNO during the seeding phase. If the seeding time window is large enough, we notice that such a cost is negligible compared to the cache miss cost. We calculate *SC* when the seeding cost is non-null in Appendix C.1, and we show that the optimal allocation is equivalent to the policy described in Section 3.3.2 when the number of requests is large.

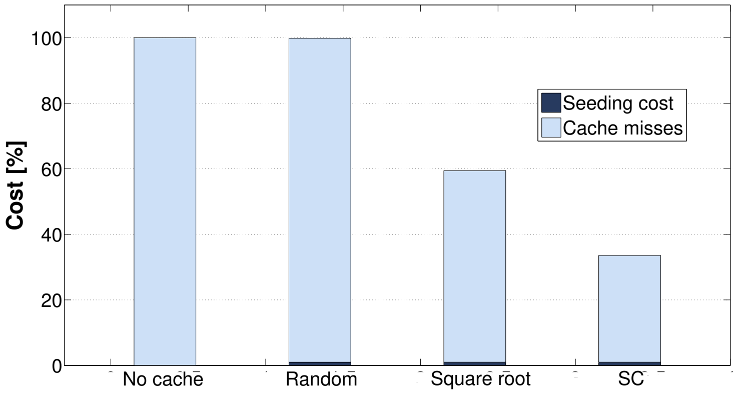

FIGURE 3.2: Percentage of traffic downloaded from the cellular infrastructure with different caching policies.

Figure 3.3 shows the cost according to the value of y_0 for the different caching policies. It is very important to note that considerable gains can be achieved with very small deadlines (in the order of a few minutes) and small number of vehicles participating in the cloud (one percent). This provides some evidence on the advantages of offloading based on a vehicular cloud compared to offloading using small cells or WiFi, as for example in Balasubramanian, Mahajan, and Venkataramani, 2010 or Lee et al., 2013: Balasubramanian, Mahajan, and Venkataramani, 2010 report minor gains for similar small deadlines, while Lee et al., 2013 require a much longer TTL (in the order of one-two hours) to achieve similar gains. In addition, increasing the deadline further has diminishing returns. This implies that even users not willing to wait too long could participate in such a system (benefiting themselves and the operator).

An efficient cache system should store as many popular contents as possible. However, in reality the catalogue of online content is really large and only a small percentage of them can be stored. Figure 3.4 shows the cost according to the buffer size. From the plot we can observe that storing 100 contents/car (only 0,1 percent of the total catalogue) provides a gain of almost 60 percent. In a scenario with a larger catalogue (e.g., 100 millions), it seems doable to store 0,1 percent of the contents (e.g., 100.000 contents/car) needed to achieve good savings. What is more, due to the intrinsic characteristics of the popularity distribution, the system might require an even smaller number of storage in order to achieve similar gains.

In an urban environment, the great availability of vehicles leads to large gains for the proposed infrastructure. However, an operator will probably keep using our framework even if the number of vehicles available decreases: in Figure 3.5 we depict the

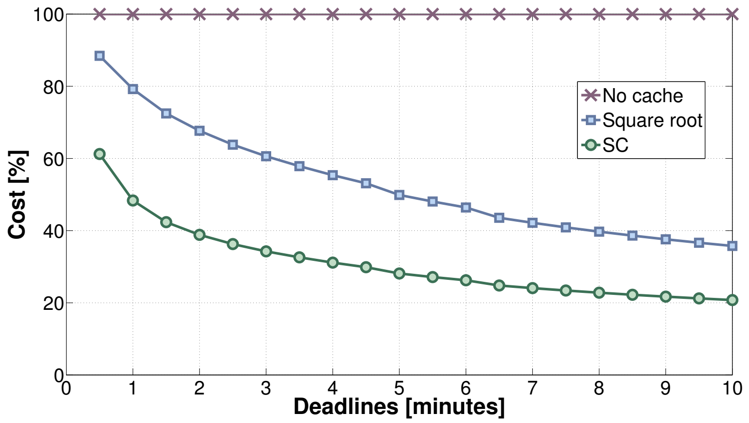

FIGURE 3.3: Percentage of traffic downloaded from the cellular infrastructure according to different deadlines y_0.

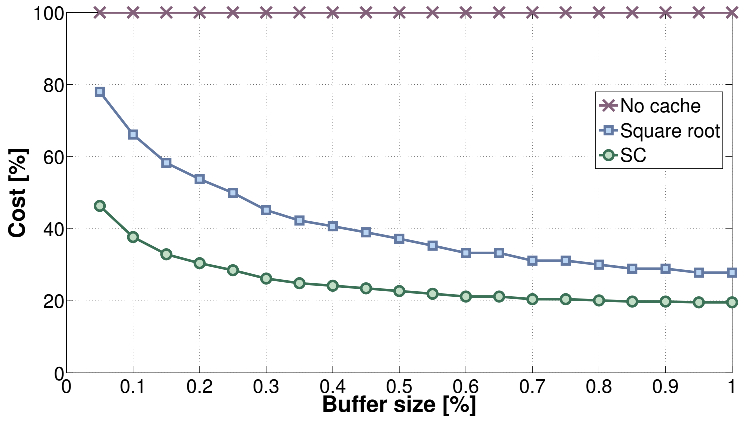

FIGURE 3.4: Percentage of traffic downloaded from the cellular infrastructure according to different buffer capacity c.

cost savings according to the number of $\mathcal{H}$ nodes participating in the cloud and the gain observed is not less than 50 percent when more than 250 vehicles are part of the cloud.

While in these simulations we have assumed that the probability of downloading a content during contact is equal to one, because of external factors, a user might not be able to get the requested content during a meeting. According to the discussion in Section 3.3.3, in Figure 3.6 we plot the percentage of savings for different values of p_i. We can note that, even when p_i is equal to $0,5$ (i.e., a $\mathcal{U}$ node loses the connection during half of the downloads[7]), *SC+* provides a gain of almost 60 percent in terms of total bytes downloaded from the core infrastructure. Clearly, this will be at the expense of some larger delay compared to the case of no disconnections. Furthermore, we plot the gain provided by *SC* (calculated with p_i equal to one) in a scenario

[7] We consider $p_i = p$ for any $i \in \mathcal{K}$ (where $p \in [0,1]$) to provide an easier interpretation of the results.

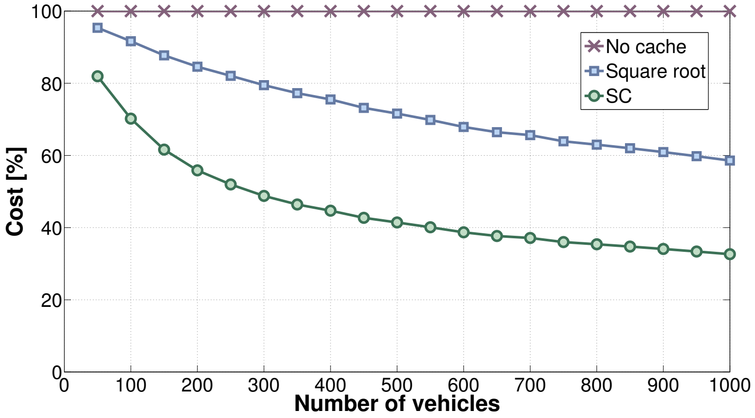

FIGURE 3.5: Percentage of traffic downloaded from the cellular infrastructure according to the number of vehicles h.

with losses: the plot shows that it is important trying to estimate the value of p_i and tune the allocation of *SC+* accordingly, since this can bring up to the 20 percent of additional savings.

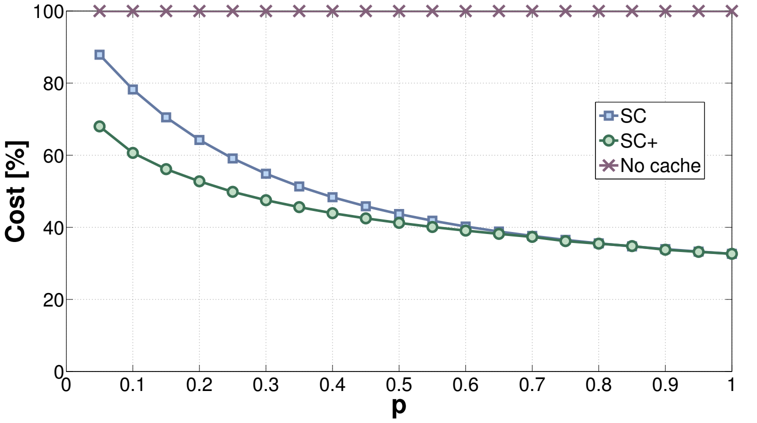

FIGURE 3.6: Percentage of traffic downloaded from the cellular infrastructure according to different values of p_i.

Finally, it has been long observed in many contexts, including Internet content, that popularity exhibits strong skewedness. To evaluate the effect of such popularity differences, in Figure 3.7 we do not take into account the real YouTube dataset, rather we consider bounded Pareto distributions (minimum value = 1 request/day, maximum value = 100.000 requests/day and ζ as shape parameter). We can observe that when the variance increases (ζ low), the optimal allocation brings a considerable gain up to the 70 percent. This is due to the fact that, if ζ is low, some contents have very high popularity, and caching them leads to a large number of cache hits. On the other hand, the gain goes to 0 when ζ increases, i.e., the differences in the content popularity are negligible making it hard to create enough cache hits with any subset of them that can fit in the cloud.

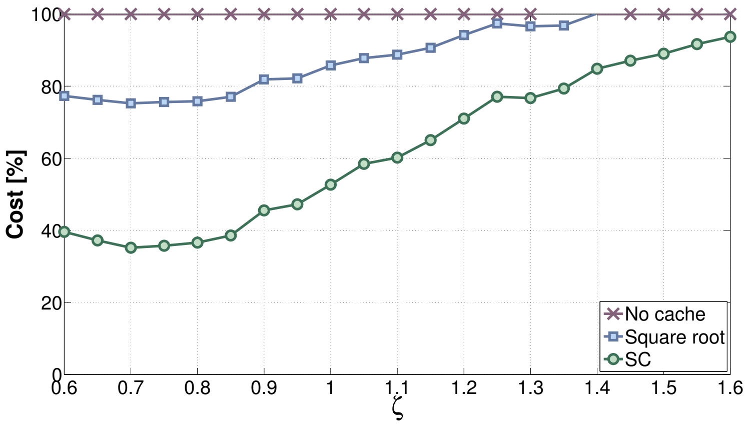

FIGURE 3.7: Percentage of traffic downloaded from the cellular infrastructure with synthetic trace for content popularity.

3.5 Summary

In this chapter, we have introduced a network infrastructure made up of vehicles storing replicas of popular content. This vehicular cloud can be used to boost the traditional cellular network. Replicas can be accessed by nearby user equipments to offload part of the mobile traffic demand. In our model, a user is willing to wait until a maximum deadline expires in order to increase the probability to have a cache hit. Such a delay is fixed and is decided by the mobile network operator. The goal of the study was to define an allocation policy to minimize the bytes downloaded from the cellular infrastructure. For this reason, we have formulated an optimization problem, and proposed the two following policies:

- *SC*. We have introduced a model where a content can be *entirely* downloaded during a single contact between a user and a vehicle. The related optimization problem is NP-hard, and we solve through KKT conditions after a continuous relaxation. The resulting allocation strategy allocates popular content proportionally to the logarithm of the content popularity.

- *SC+*. Due to interference, limited contact duration or other technology limitations, the download might fail during a meeting. We have proposed a specific policy to deal with a mean failure probability. The problem is again NP-hard, and we have shown that the related continuous relaxation is a special case of the *SC* policy.

Finally, we have performed numerical simulation to compare *SC* and *SC+* with other caching strategies. Among the others, we have noticed two positive aspects: (i) the traffic offloaded by the proposed allocation policies is significantly higher than any other policy under the assumption of the *single contact* model; (ii) vehicle mobility allows to have very small deadlines compared to similar works due to the fact that a user will see a much larger number of caches in a smaller amount of time compared to, e.g., femtocaching.

Chapter 4

Quality of Experience-Aware Content Caching

4.1 Introduction

The model discussed in Chapter 3 suffers from two main drawbacks: first, a user might be more willing to wait for a specific content than for another, and the allocation policies of Chapter 3 assign equal deadlines to the content catalogue; second, the Single Contact Model is appropriate when content is of small size, but we expect to see a performance degradation (in terms of traffic offloaded) as the content size increases (then p_i is expected to be low). In order to deal with such limitations, in this chapter we introduce three fundamental novelties:

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 our work we allow the operator to set different deadlines for different contents. This variability in the TTL brings two advantages: first, it allows to increase the percentage of the traffic offloaded through the vehicular cloud; 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. We choose to evaluate the user QoE according to the experienced *slowdown* which has become popular in recent queuing theory literature [Harchol-Balter, 2013]. 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 one-two 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 one-two 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 similar related works that use large TTLs, tuning the waiting time per content ensures maximum offloading with little QoE degradation.

Partial downloads. Due to the limited contact duration and to the content size, downloading a content in one shot might be hard, leading to a small value of p_i. Alternatively, if we consider large content such as videos, chunking is a popular way to break down the file into smaller pieces. Hence, in practice, during a contact a node could download one or more chunks. Moreover, new technologies allow easily to stop and resume the download at any time (e.g., latest versions of browsers, online music players). Later in the chapter, we introduce a model in which, when a user loses the connection, the download will resume from the point of interruption during the following cache hit.

The main focus of this chapter is on the modelling of the above scenario and on the formulation of a corresponding (nontrivial) optimization problem. The contributions of this chapter 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 (captured by the slowdown metric) and a large range of realistic conditions (e.g., content of heterogeneous size).

2. We solve the above problem presenting a variable deadline caching policy based on the Single Contact Model. Then, we generalize such a model and we propose a caching policy which takes into account partial downloads from vehicles.

3. 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 one percent 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 proposed variable deadline policies with the ones introduced in Chapter 3.

Summarizing, this chapter is structured as follows: in Section 4.2, we present the content access protocol and the main assumptions; next, in Section 4.3, we formulate an optimization problem and we propose a variable deadline policy according to the Single Contact Model; then, we generalize the model considering partial downloads, we formulate a new optimization problem, and we propose a variable deadline policy for this generic scenario in Section 4.4; finally, we validate our findings against real trace-based simulations in Section 4.5, and we conclude with a summary in Section 4.6.

4.2 System Model

The system model used in this chapter follows the model described in Section 3.2. For this reason, to avoid redundancy, we summarize briefly the common points between the two models, and we highlight the differences introduced by the user QoE-awareness.

4.2.1 Content Access Protocol

We consider the same network formed by $\mathcal{I}$, $\mathcal{H}$ and $\mathcal{U}$ nodes. The basic protocol is made up of three phases:

- ($\mathcal{I} \rightarrow \mathcal{H}$). $\mathcal{I}$ nodes place content in $\mathcal{H}$ nodes according to the chosen allocation policy (seeding phase).
- ($\mathcal{H} \rightarrow \mathcal{U}$). An end user node can request content i to the vehicles that are inside her communication range. 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 allocation policy during the seeding phase[1]. The related local access cost is assumed to be 0.
- ($\mathcal{I} \rightarrow \mathcal{U}$). 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.

4.2.2 Main Assumptions

For the sake of clarity, we summarize briefly the assumptions that have already been used in Chapter 3.

A.1 - Catalogue. Let $\mathcal{K}$ be the set of all possible contents that users might request 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 ϕ_i measured as the expected request rate within a seeding time window from all users and all cells. Content is sorted by decreasing popularity as $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_k$.

A.2 - Mobility model. We assume that the inter-meeting times T_{ij} between a user requesting content $i \in \mathcal{K}$ and a vehicle $j \in \mathcal{H}$ are IID random variables characterized

[1] 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 [Gao et al., 2016; Cai, Koprulu, and Shroff, 2013]. 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.

by a known 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 $i \in \mathcal{K}$ and *any* vehicle storing such a content.

A.3 - Cache model. Let x_i denote the number of $\mathcal{H}$ nodes storing content i. Fractional storage is not allowed.

A.4 - Content download. We assume that a content can be entirely downloaded during a contact with probability p_i. For simplicity, we first consider $p_i = 1$ for any $i \in \mathcal{K}$. However, it is trivial to extend the results of the chapter to the case $p_i < 1$ due to Lemma 3.10. We also assume that the download restarts if content is not successfully downloaded.

Later in the chapter, we generalize Assumption A.4 (Single Contact Model) by assuming that a content can be *partially* downloaded from the vehicular cloud. Then, we introduce the following assumption to deal with the user QoE:

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

$$\omega_i \triangleq \frac{y_i + t_i}{t_i} = 1 + \frac{y_i}{s_i/r},$$

where ω_i 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 the 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

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

For simplicity, we will refer to $\Omega(\mathbf{y})$ as *mean slowdown*. As we will see in Section 4.5.2, 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, the MNO can set a maximum tolerable deadline y_{max} to avoid excessively large TTLs for specific content.

The main notation used in the chapter is summarized in Table 4.1.

TABLE 4.1: Notation used in the chapter.

CONTROL VARIABLES	
x_i	Number of replicas stored for content i
X	Feasible region for $\mathbf{x}$
y_i	Deadline for content i
Y	Feasible region for $\mathbf{y}$
CONTENT	
k	Number of content in the catalogue
ϕ_i	Request rate 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
T_i	Inter-meeting time between $\mathcal{U}$ and any $\mathcal{H}$ nodes with content i
λ	Mean inter-meeting rate between $\mathcal{U}$ and $\mathcal{H}$ nodes
M_i	Number of contacts within y_i
h	Number of vehicles
CHUNK DOWNLOAD	
w_{ij}	Bytes downloaded per contact
μ	Mean of w_{ij}
σ^2	Variance of w_{ij}
W_i	Total bytes downloaded for content i from $\mathcal{H}$ nodes
f_{W_i}	Probability density function of W_i
F_{W_i}	Cumulative density function of W_i
QOE PARAMETERS	
r	Download rate from cellular infrastructure (or playout rate for videos)
Ω	Mean slowdown
y_{max}	Maximum deadline
ω_{max}	Upper bound on the mean slowdown
SETS	
$\mathcal{I}$	Infrastructure nodes
$\mathcal{H}$	Helper nodes
$\mathcal{U}$	End user nodes
$\mathcal{K}$	Content catalogue

4.3 Optimal Content Allocation with Single Contact Model

Based on the aforementioned content access protocol and on the previous assumptions, in Section 4.3.1 we formulate an optimization problem to reduce the load on the cellular infrastructure considering variable deadlines and a QoE constraint. Then, in Section 4.3.2, we propose an algorithm to solve such a problem.

4.3.1 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.

Problem 4. *Consider the Single Contact Model when deadlines are variable. The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud:*

$$\begin{aligned} \underset{\mathbf{x}\in X^k,\mathbf{y}\in Y^k}{\text{maximize}} \quad & \sum_{i=1}^{k} \Phi_{qsc} \triangleq \phi_i \cdot s_i \cdot (1 - e^{-\lambda \cdot x_i \cdot y_i}), && (3) \\ \text{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h, \\ & \Omega(\mathbf{y}) \leq \omega_{max}. && (4) \end{aligned}$$

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* $\mathbf{x}$ *and* $\mathbf{y}$.

4.3.2 QoE-Aware Content Caching with Single Contact Model (qSC)

Problem (4) 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 4.1. *Problem (4) is an NP-hard combinatorial problem.*

Proof. The problem is NP-hard since it includes MILP as a subproblem [Kannan and Monma, 1978]. □

Similarly to the previous chapter, we consider a continuous relaxation of the problem. The following proposition holds:

Proposition 4.2. *Problem (4) is a biconvex optimization problem with separable constraints.*

Proof. Eq. (3) is a twice-differentiable function on the variables $\mathbf{x}$ and $\mathbf{y}$. In order to analyze the convexity of the function, we need to examine its second partial derivatives. We refer to the matrix of the second partial derivatives of the function as the Hessian $H(\mathbf{x}, \mathbf{y})$. We can determine the concavity/convexity of a function by determining whether the determinant of the Hessian is negative or positive semidefinite: specifically, the function is convex if and only if $H(\mathbf{x}, \mathbf{y})$ is positive semidefinite for all pairs $\{\mathbf{x}, \mathbf{y}\}$. The Hessian is given by[2]:

$$H = \begin{bmatrix} \lambda^2 \cdot x_i^2 \cdot e^{-\lambda \cdot x_i \cdot y_i} & (\lambda \cdot x_i \cdot y_i - 1) \cdot e^{-\lambda \cdot x_i \cdot y_i} \\ (\lambda \cdot x_i \cdot y_i - 1) \cdot e^{-\lambda \cdot x_i \cdot y_i} & \lambda^2 \cdot y_i^2 \cdot e^{-\lambda \cdot x_i \cdot y_i}. \end{bmatrix}$$

Thus,

$$\text{det}|H| = (2 \cdot \lambda \cdot x_i \cdot y_i - 1) \cdot e^{-\lambda \cdot x_i \cdot y_i}$$

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_{qsc} 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 all linear and the feasible regions for the control variables are convex, the optimization 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 and Visweswaran, 1990. 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 2) that modifies the one described by Wendell and Hurter, 1976. 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: specifically, we use *Lagrangian relaxation* which is well suited to the solution of limited-resource allocation problems [Everett III, 1963]. Here the details of the Multi-Start Alternate Convex Search algorithm:

1. Let $\mathbf{y}^0 \in Y^k$ denote an arbitrary initial feasible set of solutions for Problem (4).

2. Solve the following convex nonlinear problem:

$$\mathbf{x}^0 \leftarrow \max_{\mathbf{x} \in X^k} \sum_{i=1}^{k} \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i \cdot y_i^0},$$
$$\text{s. t.} \quad \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h.$$

   The constraint on the maximum slowdown is implicitly satisfied by y^0. We note that this corresponds to Problem (2). Then, the solution can be easily

[2] Without loss of generality, we remove the positive constants in order to facilitate the reading.

found through KKT conditions as provided by Theorem 3.5:

$$x_i^0 = \begin{cases} 0, & \text{if } \phi_i < L_x, \\ \frac{1}{\lambda\cdot y_i^0}\cdot\ln\left(\frac{\lambda\cdot y_i^0\cdot\phi_i}{\rho_x}\right), & \text{if } L_x \le \phi_i \le U_x, \\ h, & \text{if } \phi_i > U_x, \end{cases} \tag{5}$$

where $L_x \triangleq \frac{\rho_x}{\lambda\cdot y_i^0}$, $U_x \triangleq \frac{\rho_x\cdot e^{h\cdot\lambda\cdot y_i^0}}{\lambda\cdot y_i^0}$, and ρ_x is an appropriate Lagrange multiplier.

3. The set $\mathbf{x}^0$ is then used as input for the same convex nonlinear problem optimized for $\mathbf{y} \in Y^k$.

$$\mathbf{y}^1 \leftarrow \max_{\mathbf{y}\in Y^k} \sum_{i=1}^{k} \phi_i\cdot s_i\cdot e^{-\lambda\cdot x_i^0\cdot y_i},$$
$$\text{s. t.}\quad \Omega(\mathbf{y}) \le \omega_{max}.$$

The capacity constraint is satisfied by $\mathbf{x}^0$. Since $\Omega(\mathbf{y}) \le \omega_{max}$ is linear on $\mathbf{y} \in Y^k$, we can solve this problem through KKT conditions:

$$y_i^1 = \begin{cases} 0, & \text{if } s_i < L_y, \\ \frac{1}{\lambda\cdot x_i^0}\cdot\ln\left(\frac{\lambda\cdot x_i^0\cdot s_i^2\cdot\sum_i\phi_i}{r\cdot\rho_y}\right), & \text{if } L_y \le s_i \le U_y, \\ y_{max}, & \text{if } s_i > U_y, \end{cases} \tag{6}$$

where $L_y \triangleq \sqrt{\frac{r\cdot\rho_y}{\lambda\cdot x_i^0\cdot\sum_i\phi_i}}$, $U_y \triangleq \sqrt{\frac{r\cdot\rho_y\cdot e^{y_{max}\cdot\lambda\cdot x_i^0}}{\lambda\cdot x_i^0\cdot\sum_i\phi_i}}$, and ρ_y is an appropriate Lagrange multiplier. The proof follows the proof of Theorem 3.5 where the Lagrangian function is

$$\mathcal{L}(\mathbf{y}) = \sum_{i=1}^{k} \phi_i\cdot s_i\cdot(1 - e^{-\lambda\cdot x_i^0\cdot y_i}) + \rho_y\cdot[\omega_{max} - \Omega(\mathbf{y})]\,.$$

4. If the stopping criterion (see below) is satisfied, then stop and $\{\mathbf{x}^0, \mathbf{y}^1\}$ is the solution. Otherwise, set $\mathbf{y}^0 \leftarrow \mathbf{y}^1$, and go to 2.

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 [Gorski, Pfeuffer, and Klamroth, 2007], we iterate the procedure described above for different arbitrary initial feasible sets, and we select the vectors $\{\mathbf{x}, \mathbf{y}\}$ that maximize Eq. (3). The accuracy of the solution depends on the number of iterations and on the parameter ϵ chosen to stop the search of the optimal vectors. While there is still no *theoretical* guarantee about the convergence, this version of the algorithm can reach the global optimum with large probability. Finally, the same considerations about failure probability during a download (see Section 3.3.3) also hold in this scenario, and Eqs. (5) and (6) can be recalculated accordingly. We refer to this policy as *QoE-Aware Content Caching with Single Contact Model* (qSC).

Algorithm 2 Multi-Start Convex Search Algorithm

```
Ensure: x, y
function MAIN
    max_f ← 0
    output ← $\{\emptyset, \emptyset\}$
    for i ← 1 to max_iter do
        $\mathbf{y}^0$ ← arbitrary feasible solution
        $\mathbf{x}^0$ ← see Eq. (5).
        $\mathbf{y}^1$ ← see Eq. (6).
        while $\Phi_{qsc}(\mathbf{x}^0, \mathbf{y}^1) - \Phi_{qsc}(\mathbf{x}^0, \mathbf{y}^0) > \epsilon$ do
            $\mathbf{y}^0$ ← $\mathbf{y}^1$
            $\mathbf{x}^0$ ← see Eq. (5).
            $\mathbf{y}^1$ ← see Eq. (6).
        if $\Phi_{qsc}(\mathbf{x}^0, \mathbf{y}^1) >$ max_f then
            output ← $\{\mathbf{x}^0, \mathbf{y}^1\}$
            max_f ← $\Phi_{qsc}(\mathbf{x}^0, \mathbf{y}^1)$
    return output
```

Finally, we compare Algorithm 2 with another approach that considers KKT conditions for non-convex problems. The results obtained by the two algorithms are the same in all of the scenarios tested. We describe this alternative approach in the following subsection.

KKT conditions applied to Problem (4)

The Lagrangian function associated to Problem (4) is

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = -\sum_{i=1}^{k} \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i \cdot y_i} + \rho_x \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i \right) + \rho_y \cdot (\omega_{max} - \Omega)\,, \qquad (7)$$

where ρ_x and ρ_y are appropriate Lagrangian multipliers (non-negative since the constraints are formulated as inequalities). For sake of clarity, without loss of generality, we omit the conditions on the bounds of $\mathbf{x}$ and $\mathbf{y}$. The corresponding KKT conditions are

$$\begin{cases} \rho_x \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i \right) = 0 \\ \rho_y \cdot (\omega_{max} - \Omega) = 0 \end{cases}$$

We compute the stationary points by computing the derivative of Eq. (7) with respect to x_i and y_i:

$$\frac{d\mathcal{L}(\mathbf{x}, \mathbf{y})}{dx_i} = \lambda \cdot \phi_i \cdot s_i \cdot y_i \cdot e^{-\lambda \cdot x_i \cdot y_i} - \rho_x \cdot s_i = 0,$$
$$\frac{d\mathcal{L}(\mathbf{x}, \mathbf{y})}{dy_i} = \lambda \cdot \phi_i \cdot s_i \cdot x_i \cdot e^{-\lambda \cdot x_i \cdot y_i} - \rho_y \cdot \frac{\phi_i \cdot r}{s_i} = 0.$$

that leads to

$$\begin{cases} \lambda \cdot x_i \cdot y_i = \ln\left(\frac{\lambda \cdot \phi_i \cdot s_i \cdot y_i}{\rho_x}\right) \\ s_i \cdot x_i = \frac{\rho_y}{\rho_x} \cdot \frac{\phi_i \cdot y_i \cdot r}{s_i}. \end{cases}$$

Then, the relation between ρ_x and ρ_y can be inferred from the constraints:

$$\sum_{i=1}^{k} s_i \cdot x_i = \frac{\rho_y}{\rho_x} \cdot \sum_{i=1}^{k} \frac{\phi_i \cdot y_i \cdot r}{s_i} \Rightarrow \frac{\rho_y}{\rho_x} = \frac{c \cdot h}{\omega_{max} - 1}.$$

$$s_i \cdot x_i = \frac{c \cdot h}{\omega_{max} - 1} \cdot \frac{\phi_i \cdot y_i \cdot r}{s_i}. \tag{8}$$

Finally, we infer the value of y_i from Eq. (8) and we replace in the other equation of the system. We obtain:

$$\frac{\lambda \cdot s_i}{\phi_i \cdot r} \cdot \frac{\omega_{max} - 1}{c \cdot h} \cdot x_i^2 = \ln\left(\frac{\lambda \cdot s_i}{\rho_x \cdot r} \cdot \frac{\omega_{max} - 1}{c \cdot h} \cdot x_i\right). \tag{9}$$

The solutions of the system of equations composed by Eqs. (8) and (9) plus the KKT conditions provide the set of optimal $\{\mathbf{x}, \mathbf{y}, \rho_x, \rho_y\}$ that solves Problem (4). Eq. (9) is a transcendental equation that can be easily solved by known numerical methods. This equation provides two different solutions for each content due to the non-convexity of the problem. However, although the problem is non-convex, and thus there is no theoretically guarantee of optimality, we note that a larger value both for $\mathbf{x}$ and for $\mathbf{y}$ maximizes Eq. (3). Hence, the result of this approach can be used equivalently to Algorithm 2 to solve Problem (4).

4.4 Optimal Content Allocation with Generic Contact Model

In this section, we generalize the previous content download model. We assume that the download resumes during a new contact as long as the deadline does not expire. Thus, a user will download from the cellular infrastructure *only* the remaining bytes (instead of the entire content as in the Single Contact Model). Hence, we modify Assumption A.4 as follows:

A.4 bis - Content download. Let w_{ij} be the number of bytes downloaded from content i by a $\mathcal{U}$ node during the j^{th} meeting[3]. w_{ij} are positive IID continuous random variables having equal mean μ and variance σ^2. Let further $M_i(y_i)$ (for sake of simplicity, we refer equivalently to M_i) be a point process counting the number of contacts within y_i. Then, we define

$$W_i \triangleq \sum_{j=1}^{M_i} w_{ij}$$

[3] The variable w_{ij} counts the bytes *per request*. To simplify the notation, we do not add the additional index. The same is for W_i.

as the number of bytes downloaded within y_i for content i.

Definition 4.3 (Generic Contact Model)**.** *Consider the content download model described by Assumption A.4 bis. We refer to this scenario as Generic Contact Model.*

Based on this model, in Section 4.4.1 we formulate a new optimization problem to reduce the load on the cellular infrastructure. We show that this problem is complex as it requires the knowledge of W_i which depends on the control variables $\mathbf{x}$ and $\mathbf{y}$. Then, we propose a specific policy to optimally cache content in the vehicular cloud when deadlines are variable (Section 4.4.2).

4.4.1 Offloading Optimization Problem

In the Generic Contact Model, 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 W_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 $\Phi_{gc}(\mathbf{x}, \mathbf{y})$ to be optimized:

Lemma 4.4. *Given the previous assumptions, the amount of bytes offloaded through the vehicular cloud during a seeding time window is given by:*

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

Corollary 4.5. *The objective function $\Phi_{gc}(\mathbf{x}, \mathbf{y})$ is equivalent to*

$$\Phi_{gc}(\mathbf{x}, \mathbf{y}) \equiv \sum_{i=1}^{k} \phi_i \cdot \int_0^{s_i} \left(1 - F_{W_i}(t)\right)\, dt, \tag{10}$$

where F_{W_i} is the CDF of W_i.

Proof. The objective function can be written as follows:

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

where f_{W_i} is the PDF of W_i. The first integral becomes equal to

$$s_i \cdot F_{W_i}(s_i) - \int_0^{s_i} F_{W_i}(t) dt$$

by integration by parts, while the second integral is trivially equal to

$$s_i \cdot (1 - F_{W_i}(s_i)).$$

After simplifying the null terms, we obtain Eq. (10). □

We formulate the following optimization problem:

Problem 5. *Consider the Generic Contact Model when deadlines are variable. The solution to the following optimization problem maximizes the bytes offloaded through the vehicular cloud:*

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

Solving Problem (5) requires the knowledge of F_{W_i} and, therefore, of W_i. We prove the following lemma:

Lemma 4.6. *Assume the number of vehicles participating in the vehicular cloud to be large, and the mean inter-meeting rate with such vehicles small. It holds that:*

1. *W_i approaches a compound Poisson process.*
2. *The first two moments of W_i are given by*

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

$$\textit{Var}[W_i] = (\mu^2 + \sigma^2) \cdot \lambda \cdot x_i \cdot y_i.$$

3. *The CDF of W_i is given by*

$$F_{W_i}(s_i) = 1 - \mathcal{L}^{-1}\left\{ e^{(w^*_{ij}(s)-1)\cdot\lambda\cdot x_i\cdot y_i} / s \right\}(s_i), \tag{11}$$

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

Proof. 1) Similarly to the proof of Lemma 3.2, let $\{\Gamma_{ij}(t), t > 0, j \in \mathcal{H} \mid x_{ij} = 1\}$ be x_i identical and independent renewal processes with *holding times* T_{ij} corresponding to the inter-arrival times between users and vehicles storing content i. Let further $\{\Gamma_i(t), t > 0\}$ be the superposition of these processes. The total number of contacts within the deadline y_i can be defined as

$$M_i(y_0) = \{\Gamma_i(y_i), y_i > 0\}, \qquad \forall i \in \mathcal{K}.$$

Remember that $\{\Gamma_i(t), t > 0\}$ forms a Poisson process. A Poisson process can be defined as a counting process that represents the total number of occurrences up to time t. Thus, $M_i(t)$ is again a Poisson process.

From Assumption A.4 bis, $W_i \triangleq \sum_{j=1}^{M_i} w_{ij}$. Observe that the reward (bytes downloaded) in each contact is independent of the inter-contact times, i.e., M_i and w_{ij} are independent, and w_{ij} are IID random variables with same distribution. Since M_i is a Poisson process, then W_i is a compound Poisson process.

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

$$\begin{aligned}\mathbf{E}[W_i] = \mathbf{E}\left[\sum_{j=1}^{M_i} w_{ij}\right] = \mathbf{E}\left[\mathbf{E}\left[\sum_{j=1}^{M_i} w_{ij}|M_i\right]\right] = \\ = \mathbf{E}\left[\sum_{i=1}^{M_i} \mu\right] = \mathbf{E}[M_i \cdot \mu] = \mathbf{E}[M_i] \cdot \mu,\end{aligned}$$

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 W_i, and then its variance using the total law of variance.

3) 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

$$W_i^*(s) = M_i^*(w_{ij}^*(s)),$$

where W_i^* (resp. w_{ij}^*) is the Laplace transform of W_i (resp. w_{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 3.2), we can write $W_i^*(s)$ as follows:

$$W_i^*(s) = e^{(w_{ij}^*(s)-1)\cdot\lambda\cdot x_i\cdot y_i}.$$

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, $F_{W_i}(s_i)$ corresponds to Eq. (11). □

Lemma 4.7. *The probability density function of W_i can be approximated by a normal distribution if $\mathbf{E}[M_i]$ large.*

Proof. In principle, the distribution of W_i is hard to determine. However, since in urban environments the number of contacts is expected to be considerably large, W_i can be approximated by a normal distribution [*Lecture Notes on Risk Theory*]. Also, it is possible to use other approximations, e.g., gamma translated, Edgeworth, normal power. □

All the quantities needed to solve the optimization problem are known from Lemma 4.6, and can be plugged in Eq. (10). 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 chapter.

4.4.2 QoE-Aware Content Caching with Generic Contact Model (qGC)

Problem (5) is again a MINLP and, thus, NP-hard. What is more, it 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. For this reason, we introduce a new objective function $\Phi_{qgc}(\cdot)$ that approximates Eq. (10) in order to convert the problem in a convex optimization problem, hence improving tractability.

Lemma 4.8. *The objective function of Eq. (10) can be approximated by*

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

Corollary 4.9. *Let $e \triangleq \Phi_{qgc} - \Phi_{gc}$ be the error introduced by Lemma 4.8. The following statements hold:*

1. *For a given $\mathbf{E}[W_i]$, as the content size s_i tends to 0 or becomes large, the approximation becomes exact, i.e., e tends to 0:*

$$\lim_{s_i \to 0} e = \lim_{s_i \to +\infty} e = 0.$$

2. *The error e is equal to*

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

Proof. 1) The approximation of Lemma 4.8 can be considered exact when content size is much larger or much lower than the expected amount of bytes downloaded per content i:

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

$$\lim_{s_i \to +\infty} \Phi_{qgc} = \lim_{s_i \to +\infty} \Phi_{gc} = \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}[W_i].$$

2) It is easy to see that

$$\mathbf{E}[\min\{W_i, s_i\}] = F_{W_i}(s_i) \cdot \mathbf{E}[W_i | W_i \leq s_i] + s_i \cdot (1 - F_{W_i}(s_i)). \tag{12}$$

$\mathbf{E}[W_i | W_i \leq s_i]$ corresponds to the truncated mean of W_i upper bounded by s_i. If the number of meetings within y_i is large, W_i can be considered as a normal distribution from Lemma 4.7. Thus, we can write its truncated mean as

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

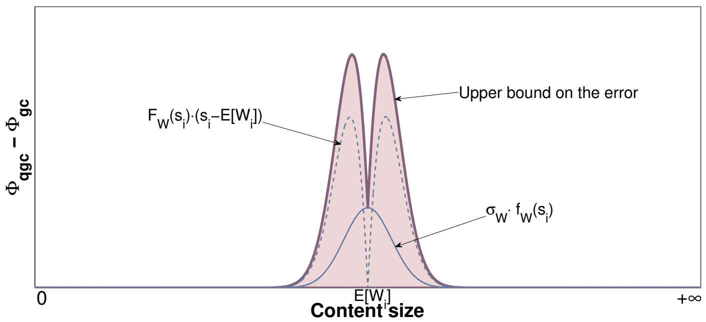

FIGURE 4.1: Error introduced by $\Phi_{qgc}(\mathbf{x}, \mathbf{y})$ in Lemma 4.8 for a fixed value of $\mathbf{E}[W_i]$.

where σ_{W_i} is the standard deviation of W_i, and can be inferred from Corollary 4.6[4]. If $\mathbf{E}[W_i]$ is larger than s_i, the error e introduced by $\Phi_{qgc}(\mathbf{x}, \mathbf{y})$ can be evaluated as follows:

$$\begin{aligned} e &= \sum_{i=1}^{k} \phi_i \cdot |\min\{\mathbf{E}[W_i], s_i\} - \mathbf{E}[\min\{W_i, s_i\}]| \\ &= \sum_{i=1}^{k} \phi_i \cdot |s_i - \mathbf{E}[\min\{W_i, s_i\}]|. \end{aligned} \tag{13}$$

Then, we compute the second term of Eq. (13) from Eq. (12), and, after some calculations, we obtain

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

Similarly, if $\mathbf{E}[W_i]$ is smaller than s_i, we have:

$$\begin{aligned} e &= \sum_{i=1}^{k} \phi_i \cdot |\mathbf{E}[W_i] - \mathbf{E}[\min\{W_i, s_i\}]| \\ &= \sum_{i=1}^{k} \phi_i \cdot [(1 - F_{W_i}(s_i)) \cdot |\mathbf{E}[W_i] - s_i| + \sigma_{W_i} \cdot f_{W_i}(s_i)] \end{aligned}$$

□

A qualitative analysis of e can be found in Fig. (4.1) where we can see that the error is concentrated in the region where $s_i \approx \mathbf{E}[W_i]$, and it tends to 0 otherwise. After a continuous relaxation, using the above approximation, Problem (5) can be converted in a *convex* optimization problem that can be solved extremely efficiently and reliably:

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

Problem 6. *Consider the approximation introduced by Lemma 4.8. Then, the solution to the following convex optimization problem maximizes the bytes offloaded through the vehicular cloud:*

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

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

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

$$\begin{aligned} \Phi_{qgc}(\mathbf{x},\mathbf{y}) &= \sum_{i=1}^{k} \phi_i \cdot \min\{\mathbf{E}[W_i], s_i\} \\ &= \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}[W_i], \quad \text{s. t. } \mathbf{E}[W_i] \leq s_i, \ \forall i \in \mathcal{K}, \end{aligned} \tag{14}$$

where the equivalence is true since the related maximization problem will choose the control variables $\mathbf{x}$ and $\mathbf{y}$ such that

$$0 \leq \mathbf{E}[W_i] \leq s_i,$$

as any scenario where $\mathbf{E}[W_i] > s_i$ is suboptimal. Remember that

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

from Lemma 4.6. According to Eq. (14), Problem (5) becomes:

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

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

[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 $c_k > 0$.

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 [Boyd and Vandenberghe, 2004]. 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 [Boyd and Vandenberghe, 2004]. □

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). We refer to this policy as *QoE-Aware Content Caching with Generic Contact Model* (qGC).

4.5 Performance Analysis

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 one percent 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 *qSC* and *qGC* with some fixed deadline policies.

4.5.1 Simulation Setup

We build a trace-driven MATLAB simulator[6] to evaluate the gain of our proposed policies. Our tool simulates YouTube video requests in the centre of San Francisco over a period of a few days. We use the following traces:

- *Vehicle mobility.* We use the Cabspotting trace [Piorkowski, Sarafijanovic-Djukic, and Grossglauser, 2009] to simulate the vehicle behaviour; this trace records the GPS coordinates for 531 taxis in San Francisco with granularity of one minute. To improve the accuracy of our simulations, we increase the granularity to 20 seconds by linear interpolation. We also use this trace to extract the

[6] While in Chapter 3 we have performed only numerical simulations, we want to highlight that in this chapter we introduce a realistic trace-driven simulator.

necessary mobility statistics for our model (e.g., the mean inter-meeting rate λ).

- *User mobility.* We use synthetic traces based on SLAW mobility model [Lee et al., 2009]. 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 [Zeni, Miorandi, and De Pellegrini, 2013]. The database includes static (e.g., title, author, duration) and dynamic information (e.g., daily and cumulative 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.

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 equal to 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 $0,1-1$ percent of the total catalogue which is an assumption that has also been used in other works [Golrezaei et al., 2013; Poularakis et al., 2014] (we use $0,2$ percent as a default value). Content size is drawn from either a gaussian or a Pareto distribution. Finally, we consider ω_{max} equal to three 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 [Coffman and Denning, 1973] which is the de facto standard in the analysis of storage systems[7]. 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 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. Finally, content requests are generated over a period of five days. Unless differently stated, we use the parameters aforementioned, summarized in Table 4.2.

We consider and compare the following allocation policies:

- *qGC.* This policy solves the optimization problem with a reasonable approximation for content of generic size. This policy is described in Section 4.4.2.

[7] The accuracy of IRM model for YouTube videos is confirmed by Crane and Sornette, 2008.

TABLE 4.2: 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}

- *qSC.* 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 4.3.2.
- *SC.* This policy solves the optimization problem when a content can be downloaded with large probability in one contact, and deadlines are fixed. This policy is described in Section 3.3.2.
- *MP.* This policy stores the most popular contents 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.* This policy allocates content randomly. Deadlines are fixed.

Maximum fixed deadline

Caching policies introduced in Chapter 3 will be used as a baseline scenario to evaluate the QoE-Aware caching policies of this chapter. In order to provide a fair comparison, we introduce the *maximum fixed deadline* y_0 as follows:

Lemma 4.10. *Let y_i be equal to y_0 for any $i \in \mathcal{K}$. The maximum value that y_0 can assume such that the QoE constraint of Eq.* (4) *is satisfied is*

$$y_0 = \frac{\omega_{max} - 1}{\sum_{i=1}^{k} \phi_i \cdot r/s_i}.$$

Proof. The value of y_0 can be directly inferred by Eq. (4) solving for $y_i = y_0$:

$$\Omega(y_0) = \sum_{i=1}^{k} \phi_i \cdot \left(1 + \frac{y_0}{s_i/r}\right) \leq \omega_{max}$$

$$\sum_{i=1}^{k} \phi_i \cdot \frac{y_0}{s_i/r} \leq \omega_{max} - 1$$

$$y_0 \leq \frac{\omega_{max} - 1}{\sum_{i=1}^{k} \phi_i \cdot r/s_i}.$$

□

4.5.2 Caching Policies Evaluation

In Figure 4.2 and Figure 4.3 we plot the amount of data offloaded for different allocation policies according to the parameters listed in Table 4.2 when mean content size is 200 MB and 50 MB. These plots also include the 95 percent confidence interval. When content size is large (Figure 4.2), the fraction of traffic offloaded by *qGC* is much larger (additional gains of around 20 percent) than any other policy in any situation. For instance, when long range communications are considered, offloading gains are in the order of 60 percent for *qGC*, and no more than 40 percent for *qSC*, *SC* and *MP*. *RAND* perform poorly in any scenario. It is also interesting to note that, while *qSC* is expected to benefit from the deadline variability, it performs similar to fixed TTL policies since the assumption that a content can be downloaded in one contact is unrealistic for content of 200 MB. On the other hand, when mean content size decreases (Figure 4.3), *qSC* becomes the best policy in any scenario. However, here *qGC* still performs better than fixed deadline policies confirming the fact that this policy reasonably approximates the generic problem even for small content. Not substantial differences have been observed for different content size distributions.

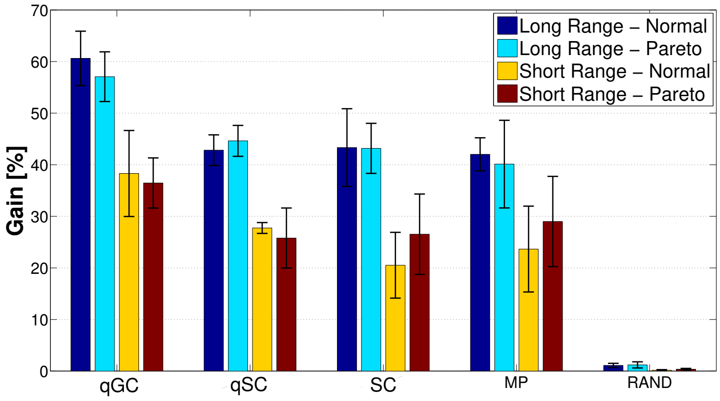

FIGURE 4.2: Percentage of traffic offloaded through the vehicular cloud when $\mathbf{E}[s]$ = 200 MB.

In Figure 4.4 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 percent of the traffic can be offloaded by *qGC*. 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 ($h = 100$), *qGC* performs poorly. This happens because the value of $\mathbf{E}[W_i]$ inferred from Lemma 4.6, that has also been used to compute *qGC*, holds only if the number of vehicles participating in the vehicular cloud is large (see Lemma 3.2). What is more, from Corollary 4.9, the error of the approximation used by *qGC* is proportional to the standard deviation of W_i which increases in a sparse environment.

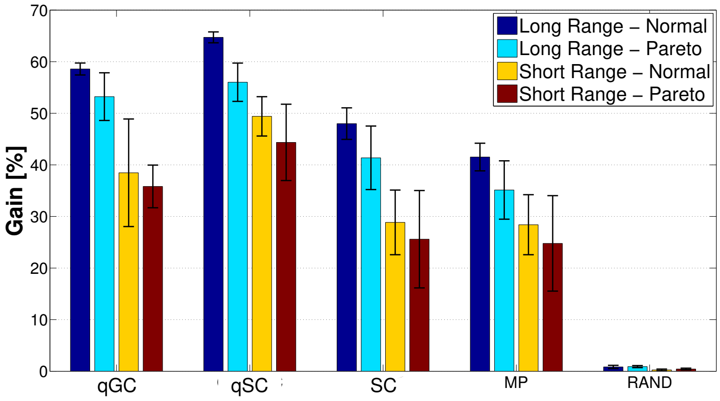

FIGURE 4.3: Percentage of traffic offloaded through the vehicular cloud when $\mathbf{E}[s]$ = 50 MB.

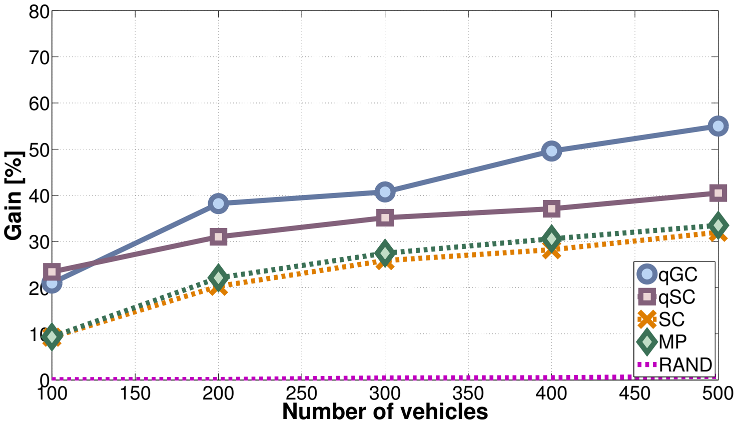

FIGURE 4.4: Percentage of traffic offloaded through the vehicular cloud according to the number of vehicles h ($c = 0,2\% \cdot k$).

Figure 4.5 compares different buffer capacities per vehicle. Buffer size goes from 0,1 to 1 percent of the catalogue (where h is equal to 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 percent of the contents needed to achieve good savings. E.g., if one considers an entire Torrent catalogue ($\sim$3 PB) or the entire Netflix catalogue ($\sim$3 PB), a mobile helper capacity of about 3 TB (0,1 percent) already suffices to offload more than 40 percent of the total traffic for long range communications (while around 30 percent for fixed deadline policies). Furthermore, as the buffer capacity increases, *qSC* offloads much more traffic than *SC*, while this is not true when the cache size per vehicle is lower than the $0,5$ percent of the catalogue. Basically, as the cache size increases, offloading gains are mainly provided by the deadline variability rather than the cache policy chosen.

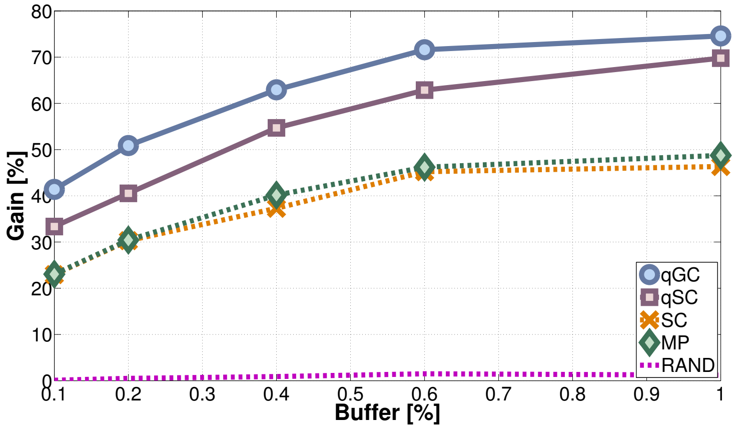

FIGURE 4.5: Percentage of traffic offloaded through the vehicular cloud according to the buffer capacity c ($h = 531$).

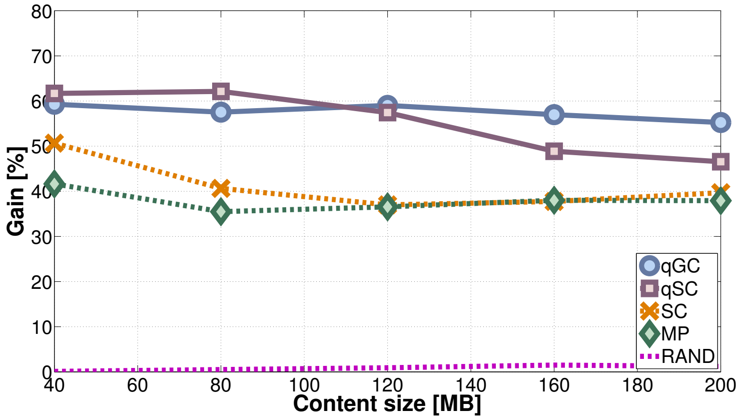

FIGURE 4.6: Percentage of traffic offloaded through the vehicular cloud according to the mean content size $\mathbf{E}[s_i]$.

In Figure 4.6 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 $\mathbf{E}[s]$ less than 80 MB), *qSC* 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 *SC* that exploits the same assumption. What is important to notice, however, is that the traffic offloaded by *qGC* is quite stable for any content size with only a slight decrease when $\mathbf{E}[s]$ is less than 50 MB.

Finally, we perform an analysis of the user QoE by allowing different values of ω_{max}. In Figure 4.7, 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 percent. We consider long range communications, and content size drawn from a gaussian distribution with mean 200 MB, but similar results can be obtained for short range communications or other content size distributions. The required mean slowdown

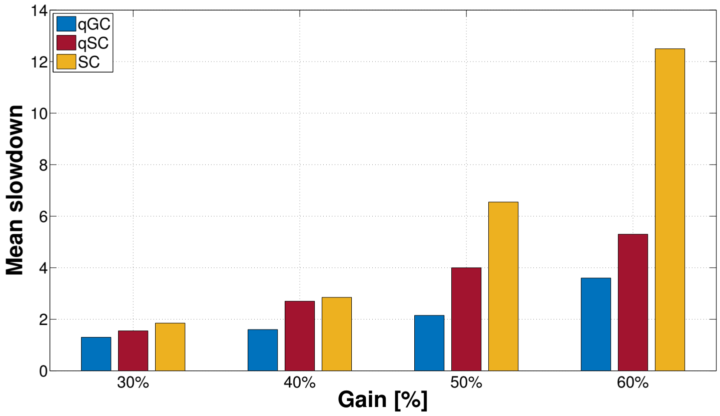

FIGURE 4.7: Mean slowdown needed to reach specific offloading gains for long range communications.

to offload more traffic increases slowly for variable deadline policies while we notice an exponential growth for fixed deadlines. Basically, Figure 4.7 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 percent increase) with *SC* policy to offload 10 percent more traffic, while the mean slowdown only increases in the range of 15-40 percent for *qGC* and *qSC* 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 thesis.

As an MNO, the policy to choose mainly depends on the characteristics of the catalogue. In general, *qGC* performs well in the majority of the scenarios. However, in sparse environments (few vehicles) the amount of bytes offloaded could drop due to the limited number of contacts. In this scenario, and in a scenario where the catalogue is made up of small content, *qSC* provides the best performance. What is more, we think that what highlighted in Figure 4.7 can be useful for an MNO to tune correctly the system parameters providing good QoE in a delay-tolerant environment which is challenging. While further approximations can be used, we believe that the QoE-aware policies proposed provide a good tradeoff between scalability and efficiency. Finally, an MNO could exploit the advantages of each policy by splitting content in two sets and use *qGC* for some, and *qSC* for others. This combined strategy would probably increase the percentage of the traffic offloaded, but we defer its theoretical study and evaluation to future work.

4.6 Summary

In this chapter, we have improved the model described in Chapter 3 by introducing three main novelties:

- *Heterogeneous deadlines*. We have let an MNO assign different deadlines to different contents. Compared to fixed deadlines, this variant brings two important advantages: (i) the user QoE degradation introduced by the delay-tolerance can be controlled by the MNO through the mean slowdown metric; (ii) the mobile traffic offloaded by the vehicular cloud increases. Hence, we have formulated a problem to jointly optimize the number of replicas to cache in vehicles *and* the deadlines to assign to each content. Such a problem is a biconvex optimization problem and we solve it with a Multi-Start Alternate Convex Search algorithm.
- *Partial downloads*. While the Single Contact Model can be successfully used when content is of small size, larger content requires a different model. For this reason, we have introduced the Generic Contact Model that allows to stop and resume the download of content chunks during different contacts. We have formulated an optimization problem that considers chunk-level downloads and we have shown that this problem is nontrivial. Through an approximation of the objective function, we were able to convert the problem in a geometric program that can be solved efficiently.
- *Trace-driven simulator*. We have built a simulator in MATLAB based on real traces for vehicle and user mobility and content popularity. The simulator considers a various number of system parameters such as download rate adaptation mechanisms and realistic cache sizes.

Finally, we have evaluated the proposed allocation policies and the impact on the user QoE. We have shown that assigning different deadlines largely improve the number of bytes offloaded through the vehicular cloud, and can also bound the QoE degradation.

Chapter 5

Content Caching for Video Streaming

5.1 Introduction

In this chapter, we exploit the fact that *streaming of stored video content offers some delay tolerance "for free"*. Video content is split into many chunks that are streamed into the user's playout buffer one by one, while consumed in parallel at the playout speed. The user does not have to wait until the whole content is found in a local cache, but she can start streaming right away, fetching chunks from the infrastructure or local/mobile caches, depending on availability of the latter and buffer status. For example, if a user is watching a one hour video, the chunks corresponding to the x^{th} minute of the video do not have to be downloaded until just before that time. During that time, a mobile cache with that chunk might be encountered and these bytes can be offloaded *without any impact on user experience*. If a node with that chunk is encountered before the user reaches that part of the video, there will be a *cache hit* for that chunk. This is in contrast with the *content download* model used in the previous chapters where the whole content must be downloaded before the user can consume it. Note that, while chunks are also "downloaded" in the streaming case, the difference here is that no extra delay needs to be *imposed* on the user (and thus deteriorate her QoE). Any chunk not available in the playout buffer when its playout time arrives, can be retrieved from main infrastructure. In this context, two interesting questions arise:

1. How many bytes of streamed video get offloaded through such a mobile edge caching?
2. How can we optimize the edge cache allocation to maximize the amount of bytes that get offloaded?

The goal of this chapter is to provide some initial answers to these questions assuming (i) vehicular nodes acting as mobile small cells and local caches, and (ii) streamed video-on-demand content as the main application[1]. Our main contributions are the following:

[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.

- We model the playout buffer dynamics of a user device in this setup as a queueing system, and analyze the expected amount of offloaded traffic (corresponding to the idle periods of this buffer) as a function of network characteristics (e.g., vehicle density, file characteristics) and a given cache allocation.
- Based on this model, we formulate the problem of optimal allocation of content in vehicles that minimizes the total load on the cellular infrastructure. We formulate the optimal allocation problem, show it is NP-hard, and propose appropriate approximations for two interesting regimes of vehicular traffic densities.
- We validate our theoretical results using real traces for content popularity and vehicle mobility, and show that our system can offload up to 50 percent of streamed data in realistic scenarios, even with modest technology penetration.

As a final remark, while we present our analysis within the context of vehicular mobile relays, the main framework and a number of our results could be applied to content streaming from fixed small cells or even user equipments (we discuss such generalizations in Section 5.5).

Summarizing, this chapter is structured as follows: in Section 5.2, we present the content access protocol and the main assumptions; next, in Section 5.3, we formulate and solve an optimization problem for different vehicular traffic regimes; then, we perform real trace-based simulations in Section 5.4, and we discuss about additional applications in Section 5.5. Finally, we conclude with a summary in Section 5.6.

5.2 System Model

Streaming video content from vehicles requires to slightly modify the system model (compared to the previous chapters). Again, to avoid redundancy, we summarize briefly the common points, and we highlight the new assumptions introduced by the streaming.

5.2.1 Video Streaming Model

We consider the same network formed by $\mathcal{U}$ nodes that request (non-live) video content for streaming to $\mathcal{I}$ or $\mathcal{H}$ 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 (at least) an $\mathcal{H}$ node that stores the requested content, the next immediate chunks not yet in the playout buffer are downloaded at low cost and *in order* at mean rate r_H. This mean rate can be easily inferred from mobility statistics (distribution of number of caches during a contact) and download rate distribution; it depends on the cell association policy used. E.g., assume that a node is currently viewing chunk n, and its playout buffer already contains chunks $n+1, \ldots, n+k$; then, chunks starting

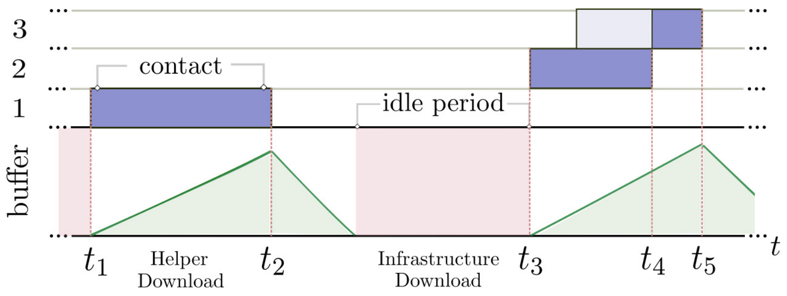

FIGURE 5.1: Sequence of contacts with three caches (above), and amount of data in end user buffer over time (below, in green). The red region indicates when data is downloaded from $\mathcal{I}$ nodes.

from chunk $n + k + 1$ will be downloaded until the connection with that $\mathcal{H}$ node is lost. This opportunistic connection is represented by the green region in Figure 5.1 (e.g., between t_1 and t_2 the node will download from cache 1). What is more, we do not allow for simultaneous connections, i.e., a $\mathcal{U}$ node can download from at most one $\mathcal{H}$ node at a time (we defer considering multi-connectivity to future work). For this reason, in Figure 5.1 the user will switch to cache 3 only at t_4, i.e., after it has finished downloading from cache 2.

- *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 $\mathcal{H}$ nodes has a high cost in terms of energy consumption [Sapountzis et al., 2014] and bandwidth of the backhaul links [Forum, 2013]. For simplicity, during a contact, our model does not allow simultaneous connections, i.e., a $\mathcal{U}$ node can download from one and only one $\mathcal{H}$ node at a time. However, our model can easily be extended to account for multiple simultaneous connections, which should provide even better offloading gains. The download from the infrastructure corresponds to the red region of Figure 5.1. However, if the playout buffer is *not* empty, no chunks are downloaded from $\mathcal{I}$ until the buffer empties.
- *Playout*. Chunks in the playout buffer are consumed at a mean viewing *playout* rate r_P.

5.2.2 Main Assumptions

We summarize briefly the assumptions that have already been used.

A.1 - Catalogue. Let $\mathcal{K}$ be the set of all possible contents that users might request 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 ϕ_i measured as the expected request rate within a seeding time window from all users and all cells. Content is sorted by decreasing popularity as $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_k$.

A.2 - Mobility model. Pairwise inter-meeting times between $\mathcal{H}$ and $\mathcal{U}$ nodes are independent, drawn from a *generic* distribution $f_C(t)$ with mean rate λ. Contact *durations* are drawn from another *generic* distribution $f_D(t)$ with mean $\mathbf{E}[D]$. We assume both distributions have bounded first and second moments.

A.3 - Cache Model. Let x_i denote the number of helper nodes storing content i. Fractional storage is not allowed.

We introduce the following new assumptions:

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 that $\epsilon = 0$). We further assume r_I and r_H to be larger than r_P in order to guarantee uninterrupted streaming. This is a reasonable assumption 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 [Hossfeld et al., 2012], and are orthogonal issues to the problem addressed in this paper. Nevertheless, our framework could be easily extended to include such initial buffering. Finally, we can consider only mean data rates since our model performs stationary regime analysis.

A.5 - Data offloading. A request for content i downloads a certain number of bytes from $\mathcal{I}$ nodes. This number is a random variable W_i that depends on x_i as well as the sample path of the contact variable(s)[2]. We denote as $\mathbf{E}[W_i|x_i]$ the expected value of this quantity, where the expectation depends on distributions $f_C(t)$ and $f_D(t)$. Our goal is to minimize it, since $s_i - \mathbf{E}[W_i|x_i]$ is the traffic *offloaded* on average for requests of content i. To keep notation simple, we will refer to this quantity as $\mathbf{E}[W_i]$.

The notation used in the chapter is summarized in Table 5.1.

5.3 Optimal Content Allocation for Video Streaming

Although the models previously described are generic and can also work for multimedia content, we note that videos have peculiar characteristics. Specifically, during the video playback, later chunks bring an "intrinsic" delay tolerance. We capture such a characteristic in a queueing theory-based model to compute the bytes offloaded through the vehicular cloud. In this section, we first formulate an optimization problem to minimize the number of bytes downloaded from the cellular infrastructure. Then, we approximate analytically this quantity for two regimes of mobile vehicle density, and we solve the related optimization problem to find the optimal content allocation. Specifically, in Section 5.3.2 we consider first a *low vehicle density* scenario which provides insights on how to solve the *generic vehicle density* scenario of Section 5.3.3. Finally, in Section 5.3.4 we provide an analytical bound on the approximation error introduced by the stationary regime analysis.

[2] Note that, differently from Chapter 4, W_i corresponds to the number of bytes downloaded from $\mathcal{I}$ nodes, instead of $\mathcal{H}$ nodes.

TABLE 5.1: Notation used in the paper.

CONTROL VARIABLES	
x_i	Number of replicas stored for content i
X	Feasible region for $\mathbf{x}$
CONTENT	
k	Number of content in the catalogue
ϕ_i	Request rate for content i
s_i	Size of content i
c	Buffer size per vehicle
MOBILITY	
λ	Mean inter-meeting rate between $\mathcal{U}$ and $\mathcal{H}$ nodes
$\mathbf{E}[D]$	Mean contact duration
h	Number of vehicles
CHUNK DOWNLOAD	
W_i	Total bytes downloaded for content i from $\mathcal{I}$ nodes
r_P	Mean viewing playout rate
r_I	Mean download rate from $\mathcal{I}$ (equal to r_P)
r_H	Mean download rate from $\mathcal{H}$
$\mathbf{E}[Y_i]$	Expected bulk size
QUEUEING PARAMETERS	
$B_i^{(n)}$	Length of the n$^{\text{th}}$ busy period of the playout buffer
$I_i^{(n)}$	Length of the n$^{\text{th}}$ idle period of the playout buffer
e_i	Error introduced by the stationary assumption for content i
SETS	
$\mathcal{I}$	Infrastructure nodes
$\mathcal{H}$	Helper nodes
$\mathcal{U}$	End user nodes
$\mathcal{K}$	Content catalogue

5.3.1 Offloading Optimization Problem

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 video plays out at the user, future parts of it are actually downloaded from locally encountered caches (in principle pre-fetched) thus offloading some of it 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 operator is to minimize the amount of bytes downloaded per content $\mathbf{E}[W_i]$, among all content $i \in \mathcal{K}$, by appropriately choosing the control variable $\mathbf{x}$. This is captured in the following:

Problem 7. *Consider the aforementioned video streaming model. The solution to the following optimization problem minimizes the expected number of bytes downloaded from the cellular infrastructure:*

$$\begin{aligned} \underset{\mathbf{x}\in X^k}{\textit{minimize}} \quad & \sum_{i=1}^{k} \phi_i \cdot \mathbf{E}[W_i], \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h, \end{aligned}$$

where $\mathbf{E}[W_i]$ is the expected number of bytes downloaded from $\mathcal{I}$ for content i when x_i helper nodes store that content.

In the limit of many content requests during a time window, the expected value of W_i becomes asymptotically exact for a specific instance of requests.

5.3.2 Video Caching with Low Density Model (VC)

First, let us assume that contacts with vehicles are sparse, i.e., the probability of overlapping contacts in time with different vehicles *both storing the same content* is small:

Definition 5.1 (Low Density Model)**.** *We refer to Low Density Model as a scenario where*

$$\lambda \cdot h \cdot \mathbf{E}[D] \ll 1 \qquad \textit{and} \qquad \frac{r_H}{r_P} < \frac{1}{\lambda \cdot h \cdot \mathbf{E}[D]}.$$

This is a reasonable assumption when the number of vehicles utilized in the cloud is small, and/or for low/medium popularity content that do not have replicas in every vehicle. In this scenario, we model the playout buffer as a bulk G^Y/D/1 queue, where new bytes arrive in bulks when a helper node with the requested content is encountered, and are consumed at a mean playout rate r_P. This system is depicted inside the small square of Figure 5.2 (the queue on the left can be ignored for now). The following holds for the expected load on the infrastructure (see Assumption

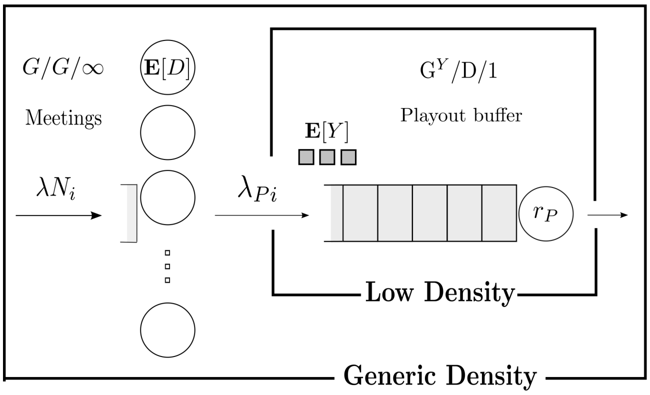

FIGURE 5.2: Proposed queuing model for the playout buffer. The queue inside the small box corresponds to the low density regime model (Section 5.3.2), while the large box (containing both queues) to the generic density regime (Section 5.3.3).

A.5):

Lemma 5.2. *Consider the Low Density Model. The following expression is asymptotically tight as the content size s_i becomes large, that is*

$$\lim_{s_i \to +\infty} \left[\mathbf{E}[W_i] - s_i \cdot \left(1 - \lambda \cdot x_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P} \right) \right] = 0.$$

Proof. Consider a content i currently stored in x_i caches. The $\mathrm{G}^Y/\mathrm{D}/1$ queue model for the playout buffer has:

- *Service Rate.* Jobs (i.e., bytes) in the buffer are served (i.e., viewed by the user) at the mean playout rate r_P.
- *Bulk Size.* A contact between a device and a helper node storing video i corresponds to a new (bulk) arrival in the playout buffer of the device. A new arrival brings a random amount of new bytes that depends on the contact duration with that $\mathcal{H}$ node. We denote the expected bulk size in bytes as $\mathbf{E}[Y_i] \triangleq \mathbf{E}[D] \cdot r_H$.
- *Arrival Rate.* The total arrival rate into the playout queue is $\lambda_{Pi} \triangleq \lambda \cdot x_i$ since there are x_i caches storing content i.

By Little's law, the long term utilization of the playout queue is

$$\rho_i = \lambda_{Pi} \cdot \frac{\mathbf{E}[Y]}{r_P} = \lambda x_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P}. \tag{15}$$

The necessary condition for this queue to be stable and ergodic is that $\rho_i < 1$, for any $i \in \mathcal{K}$. This condition is satisfied since $\lambda_{Pi} \leq \lambda \cdot x_i$ and Definition 5.1 applies.

Let $B_i^{(n)}$ (resp. $I_i^{(n)}$) be the length of the n^{th} busy (resp. idle) period of the playout buffer for content i. When the queue is stable, $\{(I_i^{(n)}, B_i^{(n)}), n \geq 1\}$ forms an alternating renewal process (as the queue regenerates at the end of each busy period). Let further $I_i^{(n)} + B_i^{(n)}$ define a cycle, and $P_I(t)$ the probability that the playout buffer is empty at time t. Since $\mathbf{E}[I_i^{(n)} + B_i^{(n)}] < +\infty$ (by stability), and the distribution of $I_i^{(n)} + B_i^{(n)}$ is non-lattice, we can apply Theorem 3.4.4 of Ross, 1983 to show that

$$\lim_{t \to +\infty} P_I(t) = \frac{\mathbf{E}[I_i]}{\mathbf{E}[I_i] + \mathbf{E}[B_i]} = 1 - \rho_i, \tag{16}$$

where the second equality holds by ergodicity. Let further associate to each cycle a reward equal to the bytes downloaded from the cellular infrastructure during that cycle, i.e., the reward in cycle n is equal to $I_i^{(n)} \cdot r_P$ (we remind the reader that the download rate from the infrastructure is assumed to be equal to the playout rate r_P, see Assumption A.4). Consider now a video of duration T_i and remember that W_i is equal to the number of total bytes downloaded from the infrastructure (see Assumption A.5). From the renewal-reward theorem (e.g., see Theorem 3.6.1 in Ross, 1983) we have that

$$\lim_{T_i \to +\infty} \frac{\mathbf{E}[W_i]}{T_i} = \frac{\mathbf{E}[I_i] \cdot r_P}{\mathbf{E}[B_i] + \mathbf{E}[I_i]}.$$

Combining Eq. (15), Eq. (16) and the fact that the duration $T_i = \frac{s_i}{r_P}$ for large s_i, we get that

$$\lim_{s_i \to +\infty} \left[\mathbf{E}[W_i] - s_i \cdot \left(1 - \lambda \cdot x_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P} \right) \right] = 0.$$

□

The above result states that as s_i becomes large, we can easily express $\mathbf{E}[W_i]$ in closed form. We will use this result as an approximation for finite size content to introduce the optimal cache allocation problem for the Low Density Model. Later, in Theorem 5.10, we elaborate on the approximation error introduced for small content.

Problem 8. *Consider the Low Density Model. The solution to the following optimization problem minimizes the expected number of bytes downloaded from the cellular infrastructure:*

$$\begin{aligned} \underset{\mathbf{x} \in X^k}{\textit{maximize}} \quad & \sum_{i=1}^{k} \phi_i \cdot s_i \cdot x_i, \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h. \end{aligned} \tag{17}$$

Proof. Our objective is to minimize the total number of bytes downloaded from $\mathcal{I}$ nodes. Based on Lemma 5.2, this is equal to

$$\underset{\mathbf{x} \in X^k}{\textit{minimize}} \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left(1 - \lambda \cdot x_i \cdot \mathbf{E}[D] \cdot \frac{r_H}{r_P} \right),$$

which is equivalent to maximize the objective function of Eq. (17). □

Proposition 5.3. *Problem (8) is an NP-hard combinatorial problem.*

Proof. The problem is a BKP with "profit" $\phi_i \cdot s_i$ and "cost" s_i for element i. BKP is NP-hard [Martello and Toth, 1990]. □

The above proposition states that the problem is hard due to the integer nature of variables x_i (by reduction to a BKP). We therefore propose the following *modified greedy* algorithm (Algorithm 3):

1. We convert the problem into a 0-1 knapsack problem using a standard transformation. We remind the reader that content is ordered in decreasing popularity (i.e., $\phi_1 \geq \phi_2 \geq \cdots \geq \phi_k$). For each content i, we create h different "virtual" contents with $\{profit, cost\}$ equal to

$$\{0,0\}, \{\phi_i \cdot s_i, s_i\}, \{2 \cdot \phi_i \cdot s_i, 2 \cdot s_i\}, \ldots, \{h \cdot \phi_i \cdot s_i, h \cdot s_i\},$$

   that gives a total of $h \cdot k$ total elements (instead of the original k).

2. We can then consider content ordered by

$$\frac{\text{profit}}{\text{cost}} = \frac{\phi_i \cdot s_i}{s_i} = \phi_i.$$

   Note that all these elements have the same profit per bit (so, for content i we can always pick the respective highest allocation $\{h \cdot \phi_i \cdot s_i, h \cdot s_i\}$, i.e., storing that content in all helpers). This corresponds to the following allocation vector:

$$x_i^0 = \begin{cases} h, & \text{if } i < \gamma \\ 0, & \text{otherwise,} \end{cases}$$

   where γ is the maximum content index such that $\sum_{i=1}^{\gamma} s_i \leq c$. This gives a greedy allocation based on profit per bit which turns out to be the content popularity ϕ_i in our case. However, it can be arbitrarily bad in some cases.

3. In order to improve the accuracy of the algorithm, we introduce the following allocation vector that replicates in any vehicle only the content with the largest profit:

$$x_i^1 = \begin{cases} h, & \text{if } i = \arg\max_{i \in \mathcal{K}} \{\phi_i \cdot s_i\} \\ 0, & \text{otherwise.} \end{cases}$$

4. We either pick $\mathbf{x}^0$ or $\mathbf{x}^1$, if the latter leads to a better total profit.

Lemma 5.4. *Algorithm 3 guarantees a $\frac{1}{2}$-approximation for the above optimization problem.*

Proof. The proof of the approximation is easy and can be found in Dantzig, 1957. □

While Algorithm 3 is $\frac{1}{2}$-optimal, it provides much better performance in realistic scenarios (as cache sizes are large and can fit several contents). Also, this algorithm stores the most popular contents in every vehicle (except in some very corner scenarios which is why we refer to it as modified). This finding is interesting because, even if contacts do not overlap, *a node still sees multiple caches during the playout of a content*. One would expect that these caches should store different content to maximize diversity. E.g., in the femtocaching setup, storing the most popular content in all caches is optimal only when caches are isolated, but it is suboptimal when a node has access to multiple caches [Golrezaei et al., 2012]. We will see that this allocation is no longer efficient when vehicle density increases.

Algorithm 3 Caching Algorithm for Low Density Model

```
Input: s, $\phi$, c, h
x ← $\emptyset$
j ← 1
while $\sum_{i=1}^{j} s_i \le c$ do
    $x_j \leftarrow h$
    $j \leftarrow j + 1$
if $\sum_{i=1}^{j-1} \phi_i \cdot s_i \cdot x_i \le \phi_j \cdot s_j \cdot x_j$ then
    x ← $\emptyset$
    $x_j \leftarrow h$
return x
```

Beyond providing performance guarantees, this policy is in line with the standard caching policies in single caches which store the most popular contents [Golrezaei et al., 2012]. While there exist some more effective algorithms to solve BKPs (e.g., Balas-Zemel, Fayard-Plateau, Martello-Toth, dynamic programming, branch-and-bound) [Martello and Toth, 1990], all of these approaches provide bad performance when applied to a large number of contents. We refer to this policy as *Video Caching with Low Density Model* (VC).

5.3.3 Video Caching with Generic Density Model (VC+)

We now consider a *busy* urban environment defined as follows:

Definition 5.5 (Generic Density Model). *We refer to Generic Density Model as a scenario where contacts with different vehicles (with the same content) might overlap, i.e., $\lambda \cdot x_i \cdot \mathbf{E}[D]$ is not small.*

If a user is downloading video i from node A, and the connection is lost (e.g., user or cache moves away), the user could just keep downloading from another node B storing i, also in range. Hence, as long as there is *at least* one cache with a copy within range (we denote this time interval with B_i), the user will keep downloading content

i at rate r_H[3]. We cannot apply the previous model directly, because when the user above switches from cache A to cache B, the contact with cache B might be already ongoing, and we are interested in the *residual* contact duration with B, which is generally different (unless contact durations are exponential). We can then model these overlapping contacts with an extra G/G/∞ queue in front of the playout queue (as shown in Figure 5.2). New vehicles arrive in the G/G/∞ queue with rate $\lambda \cdot x_i$, each staying for a random service time (corresponding to a contact duration with mean $\mathbf{E}[D]$) and independently of other cars. The number of jobs in the G/G/∞ queue is the number of vehicles concurrently within range of the user.

Hence, it is easy to see that: (i) the beginnings of busy periods of the queue on the left of Figure 5.2 correspond to new bulk arrivals in the playout buffer (queue on the right), and (ii) the mean duration of such busy periods, multiplied by r_H, corresponds to the (new) mean bulk size per arrival.

Lemma 5.6. *Consider the Generic Density Model. The bulk arrival statistics into the playout buffer are*

$$\lambda_{Pi} = \lambda \cdot x_i \cdot e^{-\lambda \mathbf{E}[D] \cdot x_i}, \tag{18}$$

$$\mathbf{E}[Y_i] = \frac{r_H}{\lambda \cdot x_i} \cdot \left(e^{\lambda \cdot \mathbf{E}[D] \cdot x_i} - 1 \right). \tag{19}$$

Proof. According to Lemma 3.2, the process formed by the inter-meeting times between a user and any vehicle storing video i approaches a Poisson process with rate $\lambda \cdot x_i$, if x_i large and λ small (due to the Palm-Khintchine theorem). Thus, the G/G/∞ queue capturing overlapping meetings (queue on the left of Figure 5.2) can be approximated by an M/G/∞ queue with arrival rate $\lambda \cdot x_i$ and mean service time $\mathbf{E}[D]$.

The probability that there are no jobs in the system (idle probability) is $e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}$ (this result is well known for M/M/∞ queue, but it also holds for generic contact durations by the insensitivity of the M/G/∞ queue [Harchol-Balter, 2013]). Furthermore, by ergodicity, it holds that[4]

$$\frac{\mathbf{E}[I_i]}{\mathbf{E}[B_i] + \mathbf{E}[I_i]} = e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}.$$

Since $\mathbf{E}[I_i] = \frac{1}{\lambda \cdot x_i}$, solving for $\mathbf{E}[B_i]$ gives us the expected busy period of the M/G/∞ queue and multiplying by r_H gives as the expected bulk size $\mathbf{E}[Y_i]$ of Eq. (19).

[3] We ignore for now interruptions from switching between nodes. Such delays can be very small (e.g., in the order of few milliseconds if vehicles are operating as LTE relays [Sesia, Toufik, and Baker, 2009]). We consider switching and association delays in the simulations.

[4] We slightly abuse notation for these idle and busy periods of the queue on the left, while in the proof of Lemma 5.2 we used them for the idle and busy period of the playout buffer (queue on the right).

Additionally, the beginnings of busy periods of the M/G/∞ queue correspond to (bulk) arrivals into the playout queue. The mean time between such arrivals is simply $\mathbf{E}[B_i] + \mathbf{E}[I_i]$. Hence, the arrival rate of bulks into the playout buffer is

$$\lambda_{Pi} \triangleq \frac{1}{\mathbf{E}[B_i] + \mathbf{E}[I_i]} = \frac{1}{\frac{e^{\lambda \cdot \mathbf{E}[D] \cdot x_i} - 1}{\lambda \cdot x_i} + \frac{1}{\lambda \cdot x_i}} = \lambda \cdot x_i \cdot e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}.$$

□

Lemma 5.7. *Consider the Generic Density Model. The following expression is asymptotically tight as the content size s_i becomes large, when $x_i < \frac{1}{\lambda \cdot \mathbf{E}[D]} \cdot \ln\left(\frac{r_H}{r_H - r_P}\right)$:*

$$\lim_{s_i \to \infty} \left[\mathbf{E}[W_i] - s_i \cdot \left[1 - \left(1 - e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}\right) \cdot \frac{r_H}{r_P}\right]\right] = 0. \tag{20}$$

Proof. $\mathbf{E}[Y_i]$ corresponds now to the expected bulk size for an arrival in the playout buffer (instead of $\mathbf{E}[D] \cdot r_H$ in the Low Density Model). Similarly to Lemma 5.2, we multiply the input rate λ_{P_i} of Eq. (18) with the bulk size $\mathbf{E}[Y_i]$ of Eq. (19), and divide by the playout rate r_P, to obtain the utilization of the playout buffer in the generic case:

$$\rho_i = \left(1 - e^{-\lambda \mathbf{E}[D] \cdot x_i}\right) \cdot \frac{r_H}{r_P}.$$

From this point on, we can follow the exact steps of the proof of Lemma 5.2, using this new ρ_i, to get the desired Eq. (20). Note, however, that unlike the Low Density Model, here the utilization of the playout buffer is lower than one when

$$x_i \geq \frac{1}{\lambda \cdot \mathbf{E}[D]} \cdot \ln\left(\frac{r_H}{r_H - r_P}\right) \triangleq \hat{h}, \tag{21}$$

where $\hat{h}$ is an upper bound on the allocation. Otherwise the queue is not stationary. Physically, this essentially means that the delivery capacity of the helper system is much higher than r_P, and (for long enough content) the infrastructure is not needed. In theory, this would make the playout queue non-stationary. In practice, this implies that we have allocated too many copies for this content, at least in our model. We will therefore use Eq. (21) as an additional constraint in the allocation problem. □

We can now formulate the optimal cache allocation problem for the generic density scenario:

Problem 9. *Consider the Generic Density Model. The solution to the following optimization problem minimizes the expected number of bytes downloaded from the cellular infrastructure:*

$$\begin{aligned} \underset{\mathbf{x} \in \hat{X}^k}{\textit{maximize}} \quad & \sum_{i=1}^{k} \phi_i \cdot s_i \cdot e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}, \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \leq c \cdot h. \end{aligned} \tag{22}$$

where $\hat{X} \triangleq \{a \in \mathbb{N} \mid 0 \leq a \leq \min\{h, \hat{h}\}\}$ *is the feasible region for the control variable* $\mathbf{x}$

Proof. Based on Lemma 5.7, the total number of bytes downloaded from $\mathcal{I}$ is equal to

$$\underset{\mathbf{x} \in X^k}{\textit{minimize}} \sum_{i=1}^{k} \phi_i \cdot s_i \cdot \left[1 - \left(1 - e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}\right) \cdot \frac{r_H}{r_P}\right],$$

which is equivalent to minimize the objective function of Eq. (22). However, we upper bound the feasible region of the control variable $\mathbf{x}$ according to Eq. (21). □

Proposition 5.8. *Problem* (9) *is an NP-hard combinatorial problem.*

Proof. The problem corresponds to the a nonlinear BKP which is NP-hard. □

Branch-and-bound algorithms have been developed for such problems, but they are not efficient when the size of the problem increases. Instead, we will consider here the continuous relaxation of Problem (9). It is easy to see that the continuous problem is convex, and we can solve it using standard KKT conditions. In addition to reduced complexity, this relaxation allows us to also derive the optimal allocation in closed form thus offering additional insights (e.g., compared to discrete approximation algorithms).

Theorem 5.9. *The solution of Problem* (9) *is given by*

$$x_i^* = \begin{cases} 0, & \textit{if } \phi_i < L, \\ \frac{1}{\lambda \cdot \mathbf{E}[D]} \cdot \ln\left(\frac{\lambda \cdot \mathbf{E}[D] \cdot \phi_i}{m_C}\right), & \textit{if } L \leq \phi_i \leq U, \\ \min\{h, \hat{h}\}, & \textit{if } \phi_i > U, \end{cases}$$

where $L \triangleq \frac{m_C}{\lambda \cdot \mathbf{E}[D]}$ *and* $U \triangleq \frac{m_C}{\lambda \cdot \mathbf{E}[D]} \cdot e^{\frac{\min\{h,\hat{h}\}}{\lambda \cdot \mathbf{E}[D]}}$, *and* m_C *is an appropriate Lagrangian multiplier.*

Proof. The above problem is clearly convex since the objective function is a sum of convex functions, the constraints are linear and the domain of the feasible solutions is convex. We solve it by KKT conditions. For such a problem, this method provides necessary and sufficient conditions for the stationary points to be optimal solutions. The KKT conditions for Problem (2) are

$$\begin{cases} l_i \cdot x_i = 0 \\ m_i \cdot (h - x_i) = 0 \\ m_C \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i\right) = 0 \end{cases}$$

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

$$\mathcal{L}(\mathbf{x}) = \sum_{i=1}^{k} \left[-\phi_i \cdot s_i \cdot e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i} + l_i \cdot x_i + m_i \cdot (h - x_i)\right] + m_C \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i\right).$$

TABLE 5.2: Estimated offloading gains of rounded allocation vs. continuous relaxation for different cache sizes (in percentage of the catalogue size).

Cache size	$0,02\%$	$0,05\%$	$0,10\%$	$0,20\%$
Rounded	33,148%	45,334%	54,959%	62,751%
Continuous	33,116%	45,323%	54,955%	62,750%

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 global solutions.

$$\frac{d\mathcal{L}(\mathbf{x})}{dx_i} = -\lambda \cdot \mathbf{E}[D] \cdot \phi_i \cdot s_i \cdot e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i} + m_i - l_i - m_C \cdot s_i = 0,$$

which gives the result of the Theorem after solving for x_i and considering the constraints. Similarly to Theorem 3.5, bounds L and U can be calculated from the constraints. We also consider the tighter upper-bound on x_i due to the stability condition for the playout buffer (according to the discussion in the proof of Lemma 5.7). □

We use randomized rounding on the content allocation of Theorem 5.9 to go back to an integer allocation. As argued earlier, the expected error is small when caches fit several contents. To validate this, in Table 5.2 we compare the objective value from our allocation to the one corresponding to the continuous solution of Theorem 5.9 (we report the percentage of traffic offloaded). As the latter is a lower bound on the optimal solution of the discrete Problem (9), the actual performance gap is upper bounded by the values shown in Table 5.2. We refer to this policy as *Video Caching with Generic Density Model* (VC+).

5.3.4 Non-stationary Playout Buffer

In order to calculate the number of bytes downloaded from the infrastructure, we have assumed that a video sees a stationary regime, regardless of its size (so that we can apply the renewal-reward theorem). In practice, video files have finite sizes. In the next theorem, we show that the predicted amount of such bytes calculated with the stationary assumption is in fact a lower bound on the actual number of bytes, which is only asymptotically tight. We also analytically derive the exact estimation error as a function of content size and scenario parameters. This quantity could perhaps be used to derive an even better estimate for the objective of our problem, and further improve performance.

Proposition 5.10. *The following statements are true:*

1. *The stationary estimate $\mathbf{E}[W_i]$ is a lower bound on the expected amount of bytes downloaded from $\mathcal{I}$ nodes for any content size s_i.*

2. *The additional expected number of bytes downloaded as a function of* $\mathbf{x}$ *is*

$$\mathbf{E}[e_i] = \frac{r_H \cdot \mathbf{E}[{B_i}^2]}{2 \cdot (\mathbf{E}[B_i] + \mathbf{E}[I_i])}, \tag{23}$$

where $\mathbf{E}[B_i]$ *(resp.* $\mathbf{E}[I_i]$*) is the mean busy (resp. idle) period of the playout buffer for content* i *and* $\mathbf{E}[B_i^2]$ *is its second moment.*

Proof. 1) The lower bound can be proved using a sample path argument. Consider the stationary process $S(t)$, counting the number of bytes in the playout buffer, at time t, for a very long file that has started streaming at time $-\infty$. Consider now a finite size file request that starts streaming at some random time t_0, and denote its playout buffer size as $S^{'}(t)$. If the request arrives during an idle period of $S(t)$, then $S^{'}(t_0) = S(t_0) = 0$ and the two sample paths are the same (*coupled*) from that point on. However, if the request arrives during a busy period of playout buffer $S(t)$, then $S^{'}(t_0) = 0$ but $S(t_0) > 0$, by definition of a busy period. Hence, the stationary file will download fewer bytes from the infrastructure in the next idle period, as it already has some to consume.

2) The error in the stationary estimate thus comes if the video request arrives during a busy period of $S(t)$. By renewal theory, this occurs with probability $\frac{\mathbf{E}[B_i]}{\mathbf{E}[B_i]+\mathbf{E}[I_i]}$. Furthermore, conditional on this event, the expected amount of bytes in the stationary playout buffer (i.e., the expected value of $S(t_0)$) is equal to the *age* of that busy period multiplied by r_H. From renewal theory and the inspection paradox, it holds that the expected age is equal to the expected *excess time* $\mathbf{E}[B_e]$, which is equal to $\frac{\mathbf{E}[B_i^2]}{2\mathbf{E}[B_i]}$, where $\mathbf{E}[B_i]$ and $\mathbf{E}[B_i^2]$ are the first and second moments of the busy periods of the playout buffer G^Y/D/1. Putting everything together, the expected error can be derived as

$$\mathbf{E}[e_i] = \frac{r_H \cdot \mathbf{E}[B_i]}{\mathbf{E}[B_i] + \mathbf{E}[I_i]} \cdot \frac{\mathbf{E}[B_i^2]}{2\mathbf{E}[B_i]},$$

which gives us Eq. (23). □

Corollary 5.11. *Assume that bulk arrivals are exponentially distributed. Then, we have*

$$\mathbf{E}[B] = -\tilde{B}'(s)|_{s=0} = \tfrac{\mathbf{E}[D]}{1-\rho_i},$$
$$\mathbf{E}[B^2] = \tilde{B}''(s)|_{s=0} = \tfrac{\mathbf{E}[D]^2}{(1-\rho_i)^3},$$

where $\tilde{B}(s)$ *is the Laplace transform of the busy periods.*

Proof. For low traffic (i.e., when the probability to have more than one job in the queue at the same time is low), the busy periods of the $M/G/\infty$ are trivially exponentially distributed. What is more, Hall, 1985 shows that under conditions of heavy traffic, busy periods are very nearly exponentially distributed as well. With this assumption, the playout queue G^Y/D/1 has arrival rate λ_B and mean bulk size $\mathbf{E}[Y_i]$ which are given by Lemma 5.6. Note that the busy periods of this queue are statistically equivalent to those of an M/G/1 queue, with the same arrival rate and mean service requirement $\mathbf{E}[D] = \frac{\mathbf{E}[Y_i]}{r_P}$. The utilization of this queue is

$$\rho_i = \left(1 - e^{-\lambda \cdot \mathbf{E}[D] \cdot x_i}\right) \cdot \frac{r_H}{r_P}.$$

We can thus calculate $\mathbf{E}[B_i]$ and $\mathbf{E}[B_i^2]$ from that M/G/1 queue instead. We can derive the Laplace transform of the busy periods of an M/G/1 queue in recursive form [Harchol-Balter, 2013] as

$$\tilde{B}(s) = \tilde{S}(s + \lambda_B - \lambda_B \cdot \tilde{B}(s)),$$

where $\tilde{S}(s)$ is the Laplace transform of the service time of the M/G/1 queue. While this recursion does not allow to invert $\tilde{B}(s)$, we can use it to calculate the moments as

$$\begin{aligned}\mathbf{E}[B] &= -\tilde{B}'(s)|_{s=0} = \tfrac{\mathbf{E}[D]}{1-\rho_i},\\ \mathbf{E}[B^2] &= -\tilde{B}''(s)|_{s=0} = \tfrac{\mathbf{E}[D]^2}{(1-\rho_i)^3}.\end{aligned}$$

□

5.4 Performance Analysis

In this section, we perform simulations based on real traces for vehicle mobility and content popularity to confirm the advantages of the vehicular cloud and to validate our theoretical results. While the simulator has already been described in Section 4.5, some modifications are needed to deal with the video streaming.

5.4.1 Simulation Setup

We build a trace driven MATLAB simulator to evaluate the offloading gains of our proposed policies. Our tool simulates YouTube video requests over a period of five days. To simulate the vehicle behaviour, we use the Cabspotting trace [Piorkowski, Sarafijanovic-Djukic, and Grossglauser, 2009] that records GPS coordinates for 531 taxis in San Francisco with granularity of one minute that we increase to 10 seconds by linear interpolation to improve the accuracy of our simulations. We use synthetic traces for user mobility based on SLAW model [Lee et al., 2009]. We infer the content popularity from a database with 100.000 YouTube videos [Zeni, Miorandi, and De Pellegrini, 2013]. We refer the interested reader to Section 4.5 for further information about the traces used by the simulator.

The simulator works as follows: first, it generates a set of content requests concentrated at day-time; inter-arrival times between two requests are exponentially distributed according to the IRM model [Coffman and Denning, 1973] that 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

TABLE 5.3: Parameters used in the simulations. The abbreviation *sr* (resp. *lr*) refers to *short range* (resp. *long range*) communications.

PARAM	VALUE	PARAM	VALUE
h	531 vehicles	c	$0,1\% \cdot k$
k	10.000 contents	$\mathbf{E}[s]$	250 MB
r_P	1 Mbps (720p)	r_H	5 Mbps
λ_{sr}	0,964 day^{-1}	λ_{lr}	2,83 day^{-1}
$\mathbf{E}[D]_{sr}$	31,23 s	$\mathbf{E}[D]_{lr}$	50,25 s

user requesting the content. Then, it allocates content in caches according to different allocation policies. For each request, the simulator reproduces the playout of the video: the end user buffer will be opportunistically filled when at least one cache storing the requested video is inside the communication range, depending on the mobility traces. Unless differently stated, we use the parameters summarized in Table 5.3. 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 the following plots.

We compare the following allocation policies:

- *VC+*. This policy allocates videos optimally with the Generic Density Model. This policy is described in Section 5.3.3.
- *VC*. This policy allocates videos optimally with the Low Density Model. This policy is described in Section 5.3.2.
- *Least Recently Used (LRU)*. Starting from a random initial allocation, it discards the least recently used item when there is a cache miss. Unlike the above two policies, LRU keeps updating the cache, and thus could incur higher traffic on the backhaul (from $\mathcal{I}$ to $\mathcal{H}$ nodes).
- *Random*. Content is randomly allocated in $\mathcal{H}$ nodes.

5.4.2 Caching Strategy Evaluation

In Figure 5.3 we vary the number of vehicles from 100 to 500. While the number of envisioned connected vehicles in the centre of San Francisco is expected to be much larger, it is really interesting to verify that a subset of them can still provide non-negligible offloading gains (more than 30 percent) which is important to promote the start up phase of the vehicular cloud. As proved in Section 5.3.2, the *VC* policy provides good performance in scenarios with low vehicle density: e.g., for h equal to 100, the offloading gain is almost equal to *VC+*. Conversely, for a larger number of vehicles (that introduce contact overlaps), the gap between the two policies becomes higher than 10 percent. Finally, we observe that the *LRU* policy underperforms both policies, in sparse scenarios, while it converges to the (worse) *VC* policy in dense ones. This is reasonable, as an *LRU* approximates a Least Frequently Used (*LFU*), i.e., storing the most popular contents when the popularity is stationary during the

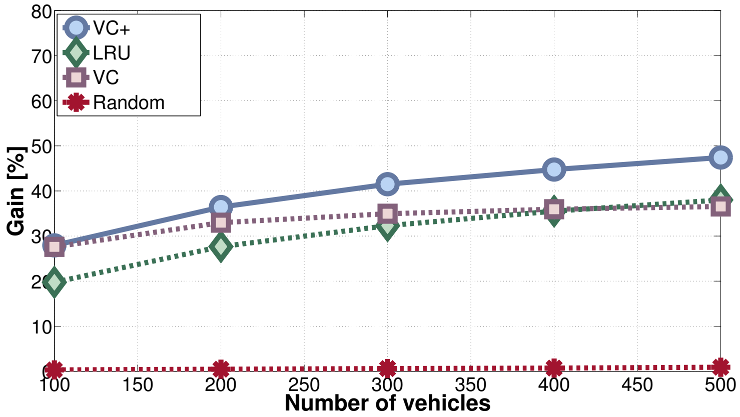

FIGURE 5.3: Percentage of traffic offloaded through the vehicular cloud according to number of vehicles h ($c = 0,1\% \cdot k$).

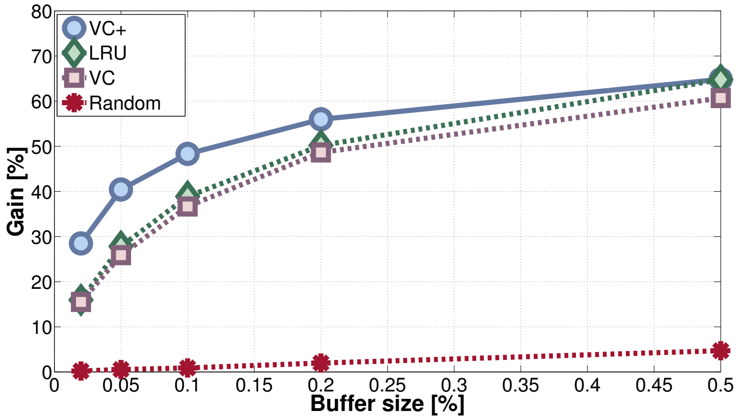

FIGURE 5.4: Percentage of traffic offloaded through the vehicular cloud according to buffer capacity c ($h = 531$).

considered window. While in some scenarios *LRU* is actually not far from the results of our allocation strategy, it should be highlighted that we ignore the extra backhaul cost due to the frequent cache updates for *LRU* (that *VC+* and *VC* do not have). In fact, simulations have shown that this "additional seeding" in *LRU* considerably degrades performance.

In Figure 5.4 we vary the cache storage per vehicle between 0,02 and 0,5 percent of the catalogue (where h is equal to 531 and mean video length is one hour). Interestingly, the smallest storage capacity still achieves considerable performance gains. E.g., if one considers an entire Torrent or Netflix catalogue (~3 PB), a mobile helper capacity of about 500 GB ($0,02$ percent) already suffices to offload 30 percent of the total traffic. Moreover, for small c (less than $0,05$ percent of the content catalogue), *VC* almost doubles the gain (from 18 to 30 percent) compared to the other policies. Finally, the *random* policy ignores the skewed Internet content popularity, and thus performs very poorly in all scenarios.

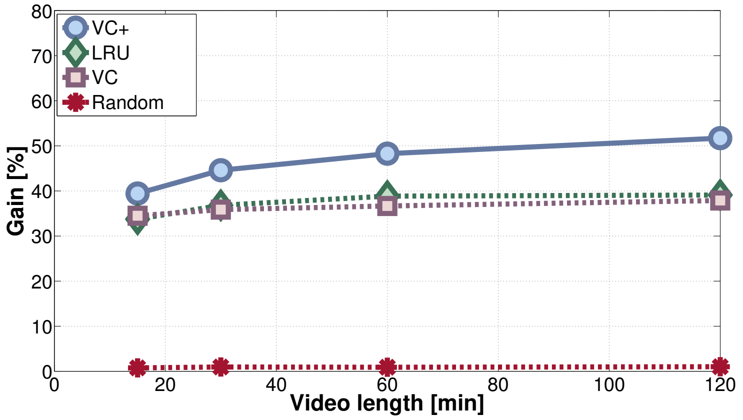

FIGURE 5.5: Percentage of traffic offloaded through the vehicular cloud according to mean video length.

Stationary regime analysis can be considered as a good approximation when there is a reasonable number of busy plus idle buffer playout periods, such that the transitory phase becomes negligible. In order to increase the number of these periods, the average content size needs to be large enough, given fixed mobility statistics. Figure 5.5 shows the fraction of data offloaded by the vehicular cloud for set of content of the same length. As proved in Proposition 5.10, gains become larger according to the average video length. However, this increase is only marginal in the majority of the scenarios: in fact, even small content (15 minutes) provides a gain which is comparable to the asymptotic gain, validating the stationary regime analysis.

5.4.3 Mobile vs. Static Helpers

In this section, we verify the pertinence of the vehicular cloud, that is based on mobile helpers, against the femtocaching framework described in Golrezaei et al., 2012, that is based on static small cells equipped with storage. In this second network, small cell helpers are distributed in the considered area proportionally to the popularity density, i.e., areas with a higher number of requests have higher small cell density (this is a common operator policy since small cells are deployed to alleviate traffic "hotspots"). Users move according to the previously described SLAW trace, and they can also download video chunks at low cost from a nearby small cell if it stores the requested video. Content is allocated using the algorithm described in Golrezaei et al., 2012. We consider two densities of small cells:

- *Femtocaching (equal number of helpers)*. From the analysis of the Cabspotting trace, the average number of vehicles *simultaneously* inside the area considered is lower than 200. In order to have a fair comparison with the vehicular cloud, we set the number of small cells to 200.
- *Low cost femto (equal cost)*. The CAPEX of a small cell consists of base station equipment, professional services (planning, installation and commissioning), and backhaul transmission equipment. This cost may range from 1000 € for a

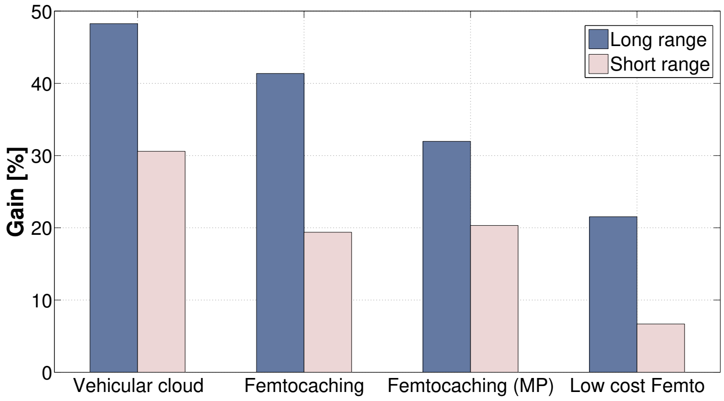

FIGURE 5.6: Percentage of traffic offloaded through the vehicular cloud and femtocaching framework.

femtocell to 20.000-30.000 € for a microcell [Senza Fili Consulting, 2013]. In the proposed vehicular cloud, the equipment might be pre-installed, and a large part of the OPEX could also be avoided as explained earlier. In fact, a first implementation of a similar vehicular cloud, where vehicles act as proxies, has shown a 10-fold cost reduction compared to small cell [*Veniam*]. We therefore also consider a sparser deployment that equalizes the total cost where we set to 50 the number of small cells[5].

Figure 5.6 compares vehicular cloud and femtocaching in terms of data offloaded. We also simulate a femtocaching scenario with the *VC* policy. As expected, gains provided by the vehicular cloud are considerably higher than femtocaching for both short and (mainly) long range communications. This result is even more interesting considering the cost: in fact, storing content in vehicles permits almost 2,5 times higher gains than femtocaching with equal cost.

5.5 Additional Use Cases

We believe one of the key strengths of our framework is its wider applicability. We present here three additional use cases:

- *Femtocaching.* In the femtocaching setup, while some of the mobility assumptions made in Section 5.2 do not exactly hold when the helpers are static and UEs are moving, our analytical formulas can still serve as a good approximation.
- *Secondary low cost operator.* Such a low cost operator could try to take up a market share by offering less expensive access to their (static or mobile) small cells to users with dual subscriptions. In this case, the optimization problem

[5] A more detailed analysis of CAPEX and OPEX can be found in Appendix A.3.

is rather for the low cost operator that aims to maximize the amount of traffic that it can serve from its own nodes.

- *Device-based caching.* More futuristic scenarios that use device-based caching and device-to-device communication as the "secondary" inexpensive network of helpers [Golrezaei et al., 2013; Han et al., 2012] could also be tackled with our framework. For example, the $\mathcal{H}$ nodes could correspond to an initial set of user equipments that the operator pushes content to. If these user equipments can further distribute the content to other user equipments (which then also become helpers) it becomes an interesting problem to transform the statistics of the size of this non-constant helper set, into the playout buffer idle statistics.

5.6 Summary

In this chapter, we have focused our study on caching multimedia content such as videos. Video streaming has become dominant in the current Internet traffic [Cisco, 2016-2021]. We have exploited the fact that later chunks introduce an intrinsic delay tolerance on the content download. Using queueing theory notions, we were able to model the intermittent contacts with the mobile caches. In this chapter, we have provided the following main contributions:

- *Modelling*. We have modelled the playout buffer at the user device with a bulk queue where arrivals correspond to the bytes opportunistically downloaded from vehicles, and the service rate corresponds to the playout video rate. Moreover, we have added an additional queue to deal with overlapping meetings, i.e., when several vehicles are inside the user's communication range at the same time.

- *Optimization*. We have calculated the number of bytes to download from the vehicular cloud based on the above queueing model. Then, we have formulated an optimization problem to infer the optimal content allocation. We have argued that, in a low density scenario, the problem corresponds to a BKP, and the optimal solution is to allocate the most popular contents in all vehicles. On the other hand, when the vehicle density increases, an optimal policy replicates content accordingly to the logarithm of the popularity.

Finally, we have improved the trace-driven simulator of Chapter 4 to validate the theoretical findings. We have also compared the vehicular cloud with the femtocaching framework [Golrezaei et al., 2012], and performed a CAPEX/OPEX analysis that has highlighted the advantages of the vehicular cloud as a function of cost reduction.

Chapter 6

Conclusion

6.1 Summary

Today's world is rapidly changing and evolving in a fully-connected environment where any electronic device is able to communicate and exchange information each other. To deal with such technology advances, a solid and widespread network infrastructure is needed. This translates into an exponential increase of mobile data demand which is predicted to double in the next five years leading to an overload of the cellular network. Researchers and industry have been proposing different solutions to deal with such a problem. Densification through small cells and caching popular content at the edge is one of the most interesting solutions because it reduces the distance between content and users and increases the spectral efficiency. In our thesis, we have exploited a futuristic network built using vehicles such as private cars, buses or taxis. The feasibility of such a vehicular network has been confirmed by several recent attempts (e.g., *Veniam*; *BMW Vehicular CrowdCell*). While there are a number of interesting technical and architectural details (which have also been discussed throughout the thesis), our main focus was on the theoretical analysis and on the modelling of such an architecture.

The main goal of the thesis was to suggest to an operator how to deal with the overload of the backhaul and the core network. Throughout the thesis, we have considered different subproblems such as the choice of the communication protocols, a system feasibility analysis, and a deep modelling of the proposed approach. What is more, we have focused our attention on the theoretical aspects of the caching strategies, while the implementation of such system is left as a future direction. Specifically, we have built a model to infer the optimal number of replicas of Internet content to store in vehicles to minimize the load on the cellular infrastructure. The optimal replication factor is the result of an optimization problem that is formulated depending on content (or chunk) download policies and content type. The thesis is written as an user satisfaction escalation as each chapter brings incremental improvements in the QoE guarantees thanks to progressive tighter bounds on the retrieval deadline of content. The main models have been discussed in the central chapters of the thesis where the major contributions lie:

- *Content caching through a vehicular cloud*. A user that requests a content agrees to wait until a maximum deadline while querying nearby vehicles to download content at low cost. We have formulated an optimization problem and inferred an optimal allocation policy to store replicas of popular content in vehicles. We have seen that such a model can be used to offload more traffic than related work while keeping deadlines much shorter due to the intrinsic vehicle mobility.
- *Quality of Experience content caching*. Starting from the idea that a user might be more willing to wait for a content of larger size, we have introduced the slowdown metric to evaluate the QoE. We have modified the previous optimization problem by including an additional "QoE budget" constraint that can be redistributed between contents. The solution of the problem was to calculate jointly the optimal replication factor *and* the deadlines to assign to the content catalogue. We have argued that the variability in the deadlines brings two main advantages compared to the fixed case: (i) the amount of traffic offloaded is much larger given the same mean slowdown; (ii) an MNO can tune the QoE constraint to have a tradeoff between traffic offloaded and user satisfaction.
- *Content caching for video streaming*. Finally, we have focused on video streaming since the majority of the mobile traffic is made up of multimedia content. We have exploited the characteristics of the video streaming to deliver content without additional delay. We have introduced a model based on queuing theory. Such a model is generic, and can be applied to any scenario with "low cost but intermittent" and "expensive but always available" sources.

In this thesis, we have assumed that a user can download a content (or part of it) while she is inside the communication range of a vehicle storing it. We have started our dissertation with the Single Contact Model where we have assumed that a content can be entirely downloaded during a contact. Then, we have generalized such a model to account for partial downloads, i.e., during a contact a user can download part of a content depending on contact duration and download rate. While more complicated from an analytical point of view, a finer-grained description of the content downloads describes better the interaction between vehicles and users. In order to validate our theoretical findings, we have performed extensive simulations based on real traces. In particular, we have argued that an optimal content allocation leads to a decrease of more than half of the total traffic load under realistic conditions.

We believe that this thesis can be considered as a *ground-breaking work* because it reveals the potential of an additional low cost infrastructure made up of vehicles. Such a new infrastructure will probably be available to everybody in the near future. Also motivated by some first attempts to define standards and physical requirements, we definitely believe that vehicles will be active part of the cellular infrastructure. Our study can be considered relevant since it provides some evidence of the potentiality of the vehicular cloud, and may speed up its adoption.

6.2 Future Work

As a future work, we suggest the following main research directions to improve and complete the work presented in the thesis:

- *Model improvements*. Increasing the number of model parameters as well as the quality of the input variables of the problem will help to build a better model. For instance, we have assumed that the popularity is uniform in a given region (and constant over a seeding time window). However, this is not necessarily true as some contents might be more popular in a certain area (e.g., football videos are more likely to be downloaded near a stadium). To deal with this, it would be useful to consider location based popularity using the recent studies on floating contents (Thompson, Crepaldi, and Kravets, 2010; Hyytiä et al., 2011). Another example where the model can be improved concerns Chapter 4: while we have shown that *qGC* performs well in the majority of the scenarios, it would be interesting to investigate a better approximation for the Generic Contact Model, e.g., considering the content type along with the content size. Also mixed policies to tie *qGC* to *qSC* can probably bring additional gains.

- *Variable video quality caching*. New adaptive streaming protocols allow to vary the video quality per chunk in an opportunistic way according to the current network conditions. However, in this work we have considered that each content has a fixed resolution. The model described in Chapter 5 can be adapted easily to variable video qualities since the potential fluctuations of the play-out rate are naturally absorbed by our queuing model that is based on average statistics. Then, further optimization can be proposed by exploiting the characteristics of such new protocols. For instance, it is possible to formulate an optimization problem to select both optimal number of copies and resolutions that minimize the load on the cellular infrastructure.

- *Per chunk caching for video streaming*. We have also assumed that the whole content is cached. However, within our video streaming framework (see Chapter 5), early chunks have a smaller chance to be downloaded from a helper node 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 most video content requested are partially watched. These two “forces” call for a per chunk optimization policy: (a) 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; (b) however, the latter suggests that early chunks have different popularity than later chunks and thus perhaps deserve more storage space. We believe these two opposing forces would lead to interesting tradeoffs when one tries to apply our framework to chunk caching rather than content caching. However, it also increases the modelling complexity, as now not every vehicle will have every chunk a user might need.

As a final remark, we think that it would also be worth trying to implement some of the communication protocols proposed to directly verify the feasibility of the system.

A real implementation of the caching strategies would definitely provide a great added value to the work.

Appendix A

Architectural details

In this section, we present some more details concerning the proposed vehicular cloud to highlight the feasibility of the approach proposed. In particular, we provide some insights about the communication protocol between mobile users and vehicles, and describe the CAPEX and OPEX to be sustained.

A.1 Communication Protocol

Previous works, such as Bychkovsky et al., 2006 and Ott and Kutscher, 2004, have confirmed the possibility to exploit opportunistic connections between vehicles and user equipments for various purposes. Such high mobility environments, which create networks with rapidly changing topologies, largely affect the choice of the communication protocol. Hadaller et al., 2007 have revealed that the two major causes for poor performance during connections (in terms of latency and bandwidth) are due to setup delays and default client bit rate selection algorithm: specifically, in order to build an architecture that guarantees good performance, the protocol should reduce setup delays with medium-long communication ranges. However, while longer ranges and high data rates improve the overall performance, battery life might be significantly lowered, especially in mobile devices.

Direct device-to-device communications between handheld devices have not been widespread until recently, with the adoption of the WiFi Direct standard. WiFi Direct facilitates setup of device-to-device networks, but one device must serve as an access point and all other devices must communicate through it, thus not supporting highly mobile networks with rapidly changing topologies [Choi et al., 2014]. What is more, WiFi Direct has been shown to be energy inefficient [Trifunovic et al., 2013].

The recent increasing interest in vehicular networks has led to the proliferation of new standards and protocols for high mobility environments [Hameed Mir and Filali, 2014]. 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) [Lin et al., 2012]. 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 [Choi et al., 2014].

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 [3GPP, 2014]. The ProSe envisages two basic functionalities:

- *ProSe discovery* that identifies the ProSe-enabled devices in proximity.
- *ProSe communication* that enables establishment of communication paths via PC5 interface between two or more ProSe-enabled devices that are in direct communication range.

The need of privacy and security has led the research to an evolved packet core-level discovery procedure, where a 3GPP network would act as a trusted intermediary and implement all of the necessary policies on behalf of users. In our scenario, the MNO will redirect the user equipment to the vehicular cloud if a new vehicle is discovered during (i): in this way, the user does not have to periodically probe nearby vehicles, resulting in a more efficient battery utilization. Once that both ProSe-enabled devices are in contact, the PC5 interface uses the sub-frames which are reserved for uplink transmission (called "sidelink" in LTE terminology) in order to minimize the hardware impact on the user equipment and especially on the power amplifier. Differently from IEEE 802.11p, LTE Advanced improves overall throughput and performance, spectrum utilization, reliability and power consumption at the cost of higher latency. However, Kim et al., 2012 show that LTE-A is capable of satisfying delay requirements for most of the vehicular applications. ProSe also support Wi-Fi Direct for the sidelink, going beyond the intrinsic setup problems due to the high mobility of vehicles.

A.2 Interference

Interference and handover issues can be solved using technological solutions such as self organizing networks (SON) and CoMP, so as to reduce OPEX and ease the commissioning processes [Alliance, 2015]. Beyond self-healing and self-optimization mechanisms, SON main feature is self-configuration: newly added base stations are self-configured in line with a "plug-and-play" paradigm. This means both connectivity establishment, and download of configuration parameters are software. Self-configuration is typically supplied as part of the software delivery with each radio cell by equipment vendors. When a new base station is introduced into the network and powered on, it gets immediately recognized and registered by the network. The neighbouring base stations then automatically adjust their technical parameters (e.g., emission power, antenna tilt) in order to provide the required coverage and capacity, and, at the same time, avoid the interference. In CoMP, data and channel state information is shared among neighbouring cellular base stations to coordinate their transmissions. CoMP techniques can effectively turn otherwise harmful intercell interference into useful signals, enabling significant power gain, channel rank advantage, and/or diversity gains to be exploited. CoMP and SON have been included in the LTE standard.

A.3 Capital and Operational Expenditures

Most operators mention costs as the main challenge for small cell deployments. One key point of the infrastructure proposed is the simplicity to turn the current infrastructure into a working vehicular cloud. The fundamental hardware components needed are the support for 802.11p (vehicle-to-mobile device) and WiFi/LTE (vehicle-to-infrastructure), and storage capacity, usually available at low cost. Sensors (e.g., GPS) can be considered as a plus to gather additional information. Basic computational capacity are required, such as an authentication system[1], and a connection manager for heterogeneous networks.

In heterogeneous networks, CAPEX consists of base station equipment, professional services (planning, installation and commissioning), and backhaul transmission equipment. Specifically, base station equipment must be low cost and fast to install, and with high capacity, which is a complex tradeoff. While macro base station CAPEX varies from 70.000 € to 150.000 € depending on number of carriers and technology used [Senza Fili Consulting, 2013; Nikolikj and Janevski, 2014], for small cells the cost goes from 1000 € for a femtocell to 30.000 € for a microcell. In the proposed vehicular cloud, the equipment will be preinstalled in vehicles [*Smartphone on Wheels*; Green, 2014] and user only need to make a subscription to join the cloud. As many studies suggest, near future vehicles will already be equipped with storage capacity and network connectivity, reducing CAPEX to only backhaul transmission equipment.

Concerning OPEX, they include site rental, electric power, maintenance, spectrum rental, and backhaul transmission lease. Furthermore, we expect wireless backhaul to dominate in small cell sites, both for cost considerations (i.e., high installation and recurring costs of fiber), and for operational considerations (i.e., difficulties in bringing fiber to structures like lampposts, and lack of flexibility). Beyond CAPEX, the vehicular cloud can also reduce OPEX since site rental and electric power are not an issue. What is more, maintenance is usually managed by the car manufacturer (e.g., periodic car inspections), limiting the costs to spectrum rental and backhaul transmission lease.

[1] Authentication is only required when vehicle caches are filled by the MNO. Communication between mobile users and vehicles do not require authentication to reduce the latency.

Key challenges of small cells are site acquisition (or rental), installation costs (professional services) and backhaul, constituting today the majority of typical outdoor small cell total cost of ownership. However, site rental and installation and maintenance costs are basically negligible (because already part of the infrastructure) in the vehicular cloud. According to Senza Fili Consulting, 2013, we estimate that our proposed architecture can reduce the costs up to 60 percent per small cell, when CAPEX and OPEX are considered. What is more, this cost estimation is a lower bound, since it does not take into account the load reduction in the backhaul link given the higher data offloading than a traditional cache in small cell.

Appendix B

Network Traffic

B.1 Content popularity

With the arrival of Web 2.0, the notion of popularity for a content can assume multiple meanings, among which number of comments, ratings or feedback. In this work, we have considered the popularity as the request rate per content, since our goal is to minimize the accesses to the cellular infrastructure. 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, video streaming popularity (e.g., YouTube) shows a quite stable behaviour, making this assumption a good approximation. What is more, it has been shown that prediction techniques based on history[1] are accurate for video contents for short-medium terms (i.e., from days to few weeks) [Cha et al., 2009]. 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.

B.2 Video Streaming

Network traffic is wide and heterogeneous. Some Internet contents present unique characteristics (e.g., gaming, voice calls or encrypted traffic), hence cannot be cached. However, the vast majority of current data traffic is represented by cacheable content (e.g., Netflix, TV series, movies, software updates). In Chapter 5, we have focused on multimedia content (e.g., videos) which represents a large percentage of the total mobile data traffic [Cisco, 2016-2021]. While these applications traditionally use HTTP, leaving HTTPS to provide end-to-end encryption for protecting sensitive information (e.g., online banking transactions, authentication), a number of new content (including video streaming) has been recently adopting HTTPS. Although this might seem an insurmountable obstacle to caching, it is actually contributing to develop new protocols to combine security and in-network operations [Naylor et al., 2015].

[1] Note that MNOs know past content popularity without incurring in privacy violations.

Video is delivered in two primary streaming mechanisms: *progressive video*, in which sections of a single file are delivered in bursts; and *adaptive video*, in which chunks of differing display quality are delivered based upon the network transport capabilities. In the progressive method, a single large file is "burst-paced" into the network, with no consideration for available bandwidth. In this scenario, the content is a single video file stored on a server and HTTP "byte-ranges" are used for seeking through the video. Each byte range may either come in the same TCP connection or a new connection. For situations in which videos are available with different display quality choices (each of which corresponds to a different bitrate), the video client (or user) manually selects a particular resolution, which corresponds to a different file on the server. The result is that two users watching the "same" video at different resolutions are actually watching different files. Once chosen, this display quality is constant (i.e., video is played at constant bitrate), regardless of the network ability to transport sufficient data to avoid video interruptions. On the other hand, adaptive video modulates the display quality based on the network available transport capacity (i.e., bandwidth). It achieves this effect by fetching "chunks" of the video: the chunk chosen is of the maximum deliverable display quality. At the beginning, the video client requests the first chunk, and starts playing it. If this chunk takes too long to deliver, then the next chunk will be requested at lower display quality; if this initial chunk delivered especially quickly, then the next chunk will be requested at a higher display quality.

Content popularity, and video on demand in particular, is responsible for two fundamental shifts in consumer behavior:

- *Higher peak bandwidth levels.* Since video content is an on "demand" application, it drives traffic when it is viewed; previously, video content was often acquired in bulk via peer-to-peer networks and then consumed later.
- *Higher subscriber sensitivity to quality.* Video has rapidly changing sights and sounds, so shifts in quality (e.g., stalls, pixelization, compression artefacts, shifts up or down in resolution, changes in frame-rate) are instantly recognized by the viewer.

From the MNO perspective, video on demand decreases network efficiency since it translates into a large amount of available but unused capacity throughout the day. During these high peaks, the network is more prone to congestion; from the viewers' perspective, congestion can very visibly manifest as degradation in video streaming quality. In this work we focus on progressive streaming, while adaptive is left as future work.

Appendix C

Extensions for *SC* policy

C.1 Non-null Seeding Cost

The *SC* policy (see Section 3.3.2) can be generalized for a scenario where the seeding cost is non-null and equal to the total number of bytes pushed in the vehicles times a positive real parameter γ less than one. A typical value for γ could be $1/\sum_{i=1}^{k}\phi_i$ to give equal weight to $(\mathcal{I} \rightarrow \mathcal{H})$ and $(\mathcal{I} \rightarrow \mathcal{U})$ communications.

Problem 10. *Consider the Single Contact Model when $p_i = 1$ for any $i \in \mathcal{K}$ and the seeding cost is non-null. The solution to the following optimization problem minimizes the bytes downloaded from the cellular infrastructure:*

$$\begin{aligned} \underset{\mathbf{x}\in X^k}{\textit{minimize}} \quad & \Phi_{seed}(\mathbf{x}) \triangleq \sum_{i=1}^{k}\left[\phi_i \cdot s_i \cdot e^{-\lambda\cdot x_i\cdot y_0} + \gamma\cdot s_i\cdot x_i\right], \\ \textit{subject to} \quad & \mathbf{s}^t \cdot \mathbf{x} \le c\cdot h. \end{aligned}$$

Similarly to Theorem 3.5, we solve the continuous relaxation of Problem (10) that gives:

Theorem 3.1. *The solution of Problem (10) is given by*

$$x_i^* = \begin{cases} 0, & \textit{if } \phi_i < L, \\ \frac{1}{\lambda\cdot y_0}\cdot \ln\left(\frac{\lambda\cdot y_0\cdot\phi_i}{\rho+\gamma}\right), & \textit{if } L \le \phi_i \le U, \\ h, & \textit{if } \phi_i > U, \end{cases}$$

where $\mathbf{x}^* \triangleq \arg\max_{\mathbf{x}\in X^k}\Phi_{seed}(\mathbf{x})$, $L \triangleq \frac{\rho}{\lambda\cdot y_0}$, $U \triangleq \frac{\rho\cdot e^{h\cdot\lambda\cdot y_0}}{\lambda\cdot y_0}$, and ρ is an appropriate Lagrange multiplier.

Proof. Problem (10) 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 KKT conditions which are

$$\begin{cases} l_i \cdot x_i = 0 \\ m_i \cdot (h - x_i) = 0 \\ \rho \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i \right) = 0 \end{cases}$$

where l_i and m_i are appropriate Lagrange multipliers related to the bounds of $\mathbf{x}$. We convert $\Phi_{seed}(\mathbf{x})$ to the standard form (from minimization to maximization) and we write the related Lagrangian function $\mathcal{L}(\mathbf{x})$:

$$\mathcal{L}(\mathbf{x}) = -\sum_{i=1}^{k} \left[\phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i \cdot y_0} + \gamma \cdot s_i \cdot x_i + l_i \cdot x_i + m_i \cdot (h - x_i) \right] \\ + \rho \cdot \left(c \cdot h - \sum_{i=1}^{k} s_i \cdot x_i \right).$$

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.

$$\frac{d\mathcal{L}(\mathbf{x})}{dx_i} = \lambda \cdot y_0 \cdot \phi_i \cdot s_i \cdot e^{-\lambda \cdot x_i \cdot y_0} - \gamma \cdot s_i + l_i - m_i - \rho \cdot s_i = 0.$$

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

$$x_i = \frac{1}{\lambda \cdot y_0} \cdot \ln \left(\frac{\lambda \cdot y_0 \cdot s_i \cdot \phi_i}{s_i \cdot \rho + \gamma \cdot s_i - l_i + m_i} \right).$$

The proof continues as in the proof of Theorem 3.5. □

When the seeding cost is null, the optimal allocation always fills the caches because any other allocation would be suboptimal. On the other hand, when the seeding cost is non-null, we can identify two regimes:

- $\rho = 0$. In this regime caches are not fulfilled. This happens because γ is too large or, equivalently, the number of requests during the seeding time window is not large enough to amortize the seeding cost. However, this scenario is *not* interesting because, in practice, the number of requests is expected to be large compared to the storage capacity of the vehicular cloud.
- $\rho > 0$. In this regime all caches are fulfilled. It is easy to see that the resulting allocation is equivalent to the one of Theorem 3.5. For this reason, assuming a large enough seeding time window, we can say that the seeding cost does not modify the optimal allocation according to the Single Contact Model.

C.2 Dynamic Adaptation to Changing Popularity

In this section, we consider a more practical setup where 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. We take into account the seeding cost to make the appropriate cache updates.

Specifically, suppose that video A and video B have the same number of views per day. Moreover, suppose that video A is very popular during the day, while video B is more popular during the night. We would like to capture this behaviour by allocating more copies for A during the day and for B during the night. Since they have the same average popularity over one day, if the seeding time window is too large (e.g., one day) the two videos will be allocated the same number of copies. On the other hand, for a small seeding time window, the cache hits due to the additional copies allocated during the interval are not large enough to amortize the cost of seeding these new copies, being perhaps removed before a newer allocation is selected in the next time window.

In such a system, seeding is *incremental*, taking into account the existing allocation, and adjusting it where necessary, depending on the potential gains predicted for a shorter time window (i.e., until the next update). We propose a simple algorithm computing the number of copies to add or remove every seeding time window for each content. This heuristic allows to have an easy implementation in practice. Specifically, every time window, we make the decision if it is convenient adding more copies for a given content. Basically, seeding one more copy provides a gain equal to the number of cache misses saved by the additional copy. The gain is given by:

$$\text{gain}_i \triangleq \phi_i \cdot s_i \cdot (1 - e^{-\lambda \cdot y_0}) \cdot e^{-\lambda \cdot y_0 \cdot x_i}.$$

Then, we sort content according to the gain that can provide and, if this is higher than the seeding cost, we add a copy (or more copies) in the cloud until the buffer is full or there no other contents to seed. On the other hand, if all the caches are full, storing new copies must follow the deletion of the less popular ones; removing one copy leads to a loss equal to the additional cache misses:

$$\text{loss}_i \triangleq \phi_i \cdot s_i \cdot (e^{\lambda \cdot y_0} - 1) \cdot e^{-\lambda \cdot y_0 \cdot x_i}.$$

Then, we select the content with the highest gain and the one with the lowest loss. If $\max(\text{gain}) - \min(\text{loss})$ is greater than one, then the switching is advantageous. We call *switching* the action taken by the $\mathcal{I}$ nodes to remove a content and replace it with another one. We recompute the gain and the loss for the contents switched and we iterate until the condition is satisfied, i.e., there is at least one advantageous switching. We add/switch the contents every time window.

Finally, we analyse how refreshing the caches affects the performance of the vehicular cloud through numerical simulations. We use the MATLAB simulator described

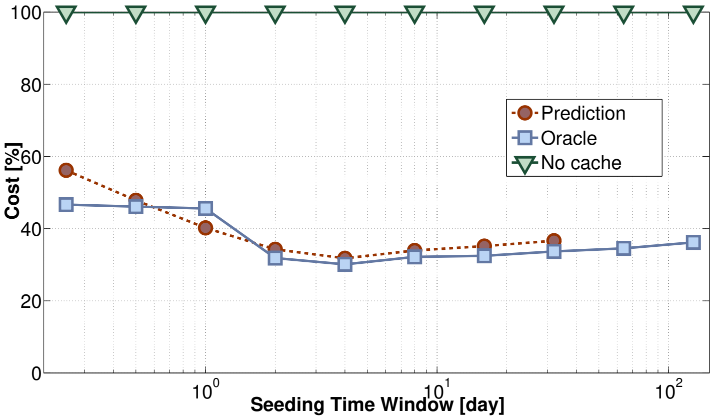

FIGURE C.1: Percentage of traffic offloaded through the vehicular cloud when a replacement policy is used.

in Section 3.4.1. Moreover, here we increase the realism of the simulations by considering that the content popularity is not known in advance, but it requires to be estimated [Dezsö et al., 2006; Gill et al., 2007]. In our simulator, we build a simple predictor to estimate the future popularity according to the previous samples, by using an exponential weighted moving average based on the latest 10 time windows. Building the best predictor goes beyond the scope of this thesis. Here, we just want to show how an error in the prediction affects the system and if the considerations done are still valid.

The content popularity provided by the database used has daily granularity; however, several studies have shown a clear sinusoidal behaviour on a daily basis [Abrahamsson and Nordmark, 2012; Gill et al., 2007]. We exploit these studies to estimate the content popularity on a hourly basis. Figure C.1 shows the final cost when caches are updated with a varying refresh time over a long time period to ensure capturing the variations in the content popularity (in the simulator, we set this long time period to one year). In this scenario, the vehicular cloud (line *prediction* in the plot) still provides a considerable number of savings (from 50 to 70 percent depending on the seeding time window). Moreover, we can have gains similar to the case with perfect knowledge (line *oracle*) even by using a simple predictor.

Appendix D

Les Véhicules comme un Mobile Cloud

D.1 Introduction

D.1.1 Le contexte

La dernière décennie a connu une croissance exponentielle du trafic mobile qui a augmenté de 63% en 2016. Le trafic de données mobiles devrait atteindre 49 exabytes par mois d'ici 2021, soit une augmentation de sept fois par rapport à 2016. Une telle demande provoque une surcharge du coeur du réseau cellulaire. Une tentative de réduire cette surcharge est donnée par la prochaine génération de technologie mobile 5G. Les chercheurs envisagent un certain nombre de nouvelles technologies de communication pour répondre à de telles exigences:

- *MIMO massif*. La technologie à antenne multiple (c'est-à-dire, MIMO) a déjà été incorporée dans LTE et Wi-Fi. Les avancées dans MIMO sont nécessaires pour utiliser un très grand nombre d'antennes pour concentrer la transmission et la réception de l'énergie du signal dans des régions plus petites.

- *Extrêmement haute fréquence*. La bande de radiofrqéuences s'étend entre 30 GHz et 300 GHz. Ce spectre de fréquence plus élevé provoque une augmentation de la bande passante qui peut être utilisée pour les communications à grande vitesse. Aujourd'hui ces fréquences sont principalement utilisées en intérieur en raison des plus faibles pertes de propagation.

- *Radio cognitive*. Cela permet à différentes technologies de partager efficacement le même spectre. Cela se fait en adaptant les canaux sans fil non utilisés et en adaptant le schéma de transmission.

- *Petites cellules*. La taille des cellules a progressivement diminué dans les zones urbaines. La densification par petites cellules promet d'améliorer l'efficacité spectrale à bon marché. Toutefois, l'introduction d'un grand nombre de petites cellules nécessite d'importantes modifications du réseau de backhaul.

Alors que le 5G pourrait offrir de bonnes performances réduisant la congestion, ces mises à jour sont assez coûteuses. Plusieurs travaux ont proposé de stocker les contenus (fichiers ou vidéos) dans les petites cellules afin de diminuer la charge sur l'infrastructure cellulaire sans surcharger le réseau de backhaul. Stocker les contenus dans les stations de base, dans les petites cellules ou dans les appareils mobiles porte un certain nombre d'avantages pour la performance du réseau comme, par example, la réduction de la latence.

D.1.2 Le problème

Le stockage des contenus peut réduire la charge sur l'infrastructure cellulaire, mais comporte également des inconvénients majeurs. Par exemple, en ce qui concerne le stockage dans les petites cellules, ils existent deux problèmes principaux: (i) une couverture étendue des petites cellules est nécessaire pour assurer une décharge du trafic des macrocellules; (ii) les évidents avantages du stockage sont encore à démontrer et des études initiales sont pessimistes. D'autre part, le stockage des contenus dans les téléphones mobiles et l'utilisation de communications locales entre périphériques est une solution à faible coût, mais cette solution est concernée par des problèmes importants comme la limitée capacité de stockage et contraintes énérgetiques.

Comme nous l'avons déjà expliqué, le réseau cellulaire actuel est surchargé par la grande demande de trafic mobile. Par conséquent, l'objectif principal de cette thèse est de suggérer à un opérateur comment réduire la charge sur l'infrastructure cellulaire grâce au stockage des contenues populaires. La solution proposée doit répondre à trois exigences fondamentales:

- *Limitation de coûts*. Les opérateurs de réseaux mobiles auraient des coûts significativement plus élevés pour 5G car ils ont besoin de déployer des milliers de petites cellules liées à des connexions à large bande passante. Ainsi, la réduction des coûts et de la consommation d'énergie sont nécessaires pour adopter des technologies de nouvelle génération.
- *Capacité de stockage*. En raison de l'immensité du catalogue Internet, une grande capacité de stockage est nécessaire pour assurer un grand nombre de *cache hit*.
- *Qualité d'expérience de l'utilisateur*. Dans le contexte des télécommunications, la qualité d'expérience est définie comme le degré de plaisir ou d'agacement de l'utilisateur d'une application ou d'un service. Une nouvelle technologie de réseau doit nécessairement tenir compte de l'impact qu'une telle solution aurait sur les utilisateurs.

Bien qu'il existe un certain nombre de défis techniques à considérer (concernant la mise en œuvre et les protocoles), ce travail est principalement axé sur l'étude analytique, la modélisation et l'optimisation.

D.1.3 Mes contributions

Afin de traiter les exigences précitées, nous proposons d'utiliser des transports privés (par exemple, des voitures) et/ou public (par exemple, des taxis, des autobus) agissant comme petites cellules mobiles où stocker des contenus populaires. L'ensemble des véhicules participant au mécanisme de déchargement forme un *cloud de véhicules*. Dans l'infrastructure proposée, un operateur mobile garde les contenus dans le cloud de véhicules pour décharger une partie du trafic. Contrairement aux déploiements de petites cellules, le cloud de véhicules apporte trois avantages fondamentaux:

- *La mobilité des véhicules étend la taille du stockage local accessible*. Avec des caches fixes, la quantité de données déchargées dépend (presque exclusivement) de la couverture femto ou picocellulaire puisque la plupart des utilisateurs présentent un comportement nomade, restant dans le même endroit pendant de longues périodes. D'autre part, un utilisateur rencontrera plusieurs véhicules pendant le téléchargement du contenu, en particulier dans un environnement urbain dense, étendant ainsi pratiquement la taille du stockage local accessible.

- *Le cloud de véhicules réduit les coûts*. Le reseau cellulaire actuelle peut facilement être transformée en un cloud de véhicules. Les véhicules ont la necessité de garantir les communications avec l'infrastructure et la capacité de stockage qui est généralement disponible à faible coût.

- *Le cloud de véhicules ouvre le marché à nouveaux operateurs mobiles*. La simplicité de transformer les villes modernes en un cloud de véhicules peut inciter des nouveaux opérateurs cellulaires virtuels à entrer sur le marché sans avoir besoin de gros investissements. Cela peut être utile, par exemple, dans les pays en développement où la demande de données augmente au même rythme que dans les pays développés.

Alors que le nombre de voitures avec une sorte de capacité de communication aujourd'hui est faible, on estime qu'environ 90% des nouveaux modèles devraient avoir une connectivité Internet d'ici 2020. Les opérateurs mobiles voient la voiture comme un autre appareil connectés à leurs réseaux et ils ont commencé à proposer des plans de données dédiés aux véhicules. Nous considérons le cloud de véhicules comme une fonctionnalité supplémentaire: lorsqu'un utilisateur navigue sur Internet, l'operateur peut décider de rediriger les requêtes aux véhicules si l'utilisateur est abonné à la fonctionnalité du cloud de véhicules. Dans ce travail, nous exploitons un tel cloud de véhicules pour stocker les contenus afin de maximiser la demande mobile déchargée. Nous construisons un modèle dans lequel un utilisateur peut télécharger de manière opportuniste un contenu ou une partie de celui-ci (c'est-à-dire des morceaux). Pour faire face à la qualité d'expérience, nous fournissons différentes stratégies d'allocation en fonction d'un niveau croissant d'expérience utilisateur perçu et d'une granularité plus fine du modèle de téléchargement des contenus. Nous énumérons ici les principales contributions du travail:

- *La modélisation*. Nous modélisons le cloud de véhicules et la communication avec les utilisateurs finaux. Une telle interaction est difficile en raison de la mobilité intrinsèque des noeuds et de leur disponibilité intermittente. Nous utilisons deux modèles pour traiter le téléchargement des contenus: d'abord, nous introduisons une granularité à grain grossier où un contenu est entièrement téléchargé (avec une certaine probabilité) pendant un seul contact; bien que cela puisse être considéré comme raisonnable pour des petits contenus, grands contenus nécessite une granularité plus fine, et nous présentons un modèle amélioré pour considérer les téléchargements au niveau du morceau.

- *Optimisation*. Selon différents modèles de téléchargement de contenu et types, nous formulons analytiquement des problèmes d'optimisation pour calculer le nombre d'éléments à allouer dans le cloud de véhicules afin de maximiser la quantité de données déchargées. Nous montrons la complexité de ces problèmes et nous proposons des heuristiques, des relaxations ou des approximations de les résoudre efficacement.
- *Analyse des performances*. Nous développons un simulateur MATLAB basées sur des traces réelles afin de soutenir nos résultats théoriques. Dans le simulateur nous considérons un certain nombre de paramètres tels que la mobilité des utilisateurs, les tailles de cache et le mécanisme d'installation et d'association pendant le téléchargement.

D.2 Résumé de la thèse

D.2.1 Stockage des contenus dans un cloud de véhicules

Nous effectuons une étude préliminaire du cloud de véhicules avec deux idées principales: (i) les véhicules sont plus étandus et nécessitent des coûts inférieurs par rapport aux petites cellules; (ii) en combinant la mobilité des véhicules avec un accès au contenu différé, il est possible d'augmenter le nombre de cache hit. Contrairement aux petites cellules fixes, lorsque les caches sont sur les véhicules, un utilisateur statique ou qui se déplace lentement rencontrera un nombre beaucoup plus élevé de caches dans le même délai. Ainsi, dans notre système, les retards maximaux sont garantis jusqu'à quelques minutes. Au-delà de cette limite, le contenu est téléchargé de l'infrastructure cellulaire. Nous proposons un model pour calculer le nombre optimal de contenus à stocker dans les véhicules pour minimiser la charge du coeur du réseau cellulaire. Nous supposons qu'un contenu peut être entièrement téléchargé pendant un seul contact, et nous présentons deux stratégies de mise en cache (*SC* et *SC*+).

Les travail relatif à ce chapitre est publié dans:

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Storage on wheels: Offloading popular contents through a vehicular cloud". In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2016, pp. 1–9. DOI: 10.1109/WoWMoM.2016.7523506

D.2.2 Stockage des contenus avec qualité d'expérience

Nous proposons à l'opérateur de fixer des délais différents pour différents contenus. L'optimisation du temps d'attente par contenu assure un déchargement maximal avec une faible détérioration de la qualité d'expérience que nous évaluons en fonction du *slowdown* expérimenté qui rapporte le délai d'attente avec le temps de téléchargement "net". Nous modélisons un tel scénario et nous formulons un problème d'optimisation pour maximiser le trafic déchargé tout en garantissant des garanties d'expérience. Nous proposons deux stratégies de délai variable avec téléchargement de fichiers au niveau du contenu (*qSC*) ou au niveau du morceau (*qGC*).

Les travaux relatifs à ce chapitre sont publiés dans:

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud". In: MSWiM (2017), *under review*

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud". In: *IEEE Transactions on Mobile Computing* (2017), *under review*

D.2.3 Stockage des contenus pour la diffusion vidéo

La grande majorité du trafic Internet concerne les vidéos. Des nouveaux services de diffusion ont été récemment introduits sur le marché (Netflix, Amazon Prime). Nous soutenons que les véhicules peuvent être utilisées pour la diffusion de vidéos à faible coût sans qu'il soit nécessaire d'imposer du retard à l'utilisateur. Les utilisateurs peuvent télécharger des morceaux de vidéo à partir des véhicules rencontrés (à faible coût) ou de l'infrastructure cellulaire (à coût élevé) lorsque leurs *buffer* sont vides tout en regardant le contenu. Nous modélisons la dynamique du buffer en tant que système de files d'attente et analysons les caractéristiques de ses périodes *idle* (pendant lesquelles l'accès à l'infrastructure cellulaire est requis). Sur la base de ce modèle, nous formulons le problème de l'allocation optimale de contenus pour minimiser la charge totale sur l'infrastructure cellulaire. Nous résolvons un tel problème d'optimisation pour differents densités de véhicules (*VC* et *VC+*).

Les travaux relatifs à ce chapitre sont publiés dans:

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Streaming Content from a Vehicular Cloud". In: *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. CHANTS '16. New York City, New York: ACM, 2016, pp. 39–44. ISBN: 978-1-4503-4256-8. DOI: 10.1145/2979683.2979684. URL: http://doi.acm.org/10.1145/2979683.2979684

Luigi Vigneri, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Low Cost Video Streaming through Mobile Edge Caching: Modeling and Optimization". In: *IEEE Transactions on Mobile Computing* (2017), *under review*

D.3 Conclusions

Le monde d'aujourd'hui change rapidement et évolue dans un environnement entièrement connecté où tous les dispositifs électroniques sont capables de communiquer et d'échanger des informations les uns des autres. Cela se traduit en une augmentation exponentielle de la demande de données mobiles. Les chercheurs et l'industrie proposent différentes solutions pour résoudre un tel problème. La densification à travers de petites cellules et le stockage des contenus populaires est l'une des solutions les plus intéressantes car elle réduit la distance entre le contenu et les utilisateurs et augmente l'efficacité spectrale. Dans notre thèse, nous avons exploité un réseau futuriste construit en utilisant des véhicules comme des petites cellules.

L'objectif principal de la thèse était de suggérer à un opérateur comment faire face à la surcharge du backhaul et du réseau de base. Tout au long de la thèse, nous avons examiné différents sous-problèmes tels que le choix des protocoles de communication, une analyse de faisabilité du système et une modélisation approfondie de l'approche proposée. De plus, nous avons concentré notre attention sur les aspects théoriques des stratégies de mise en cache, alors que la mise en œuvre d'un tel système est laissée comme une direction future. Plus précisément, nous avons construit un modèle pour inférer le nombre optimal de répliques des contenus Internet à stocker dans les véhicules afin de minimiser la charge sur l'infrastructure cellulaire. Le facteur de réplication optimal est le résultat d'un problème d'optimisation qui est formulé en fonction des règles de téléchargement des contenus (ou morceaux) et du type des contenus.

Dans cette thèse, nous avons supposé qu'un utilisateur peut télécharger un contenu (ou une partie de celui-ci) alors qu'il est proche d'un véhicule qui le stocke. Au début de notre dissertation nous avons supposé qu'un contenu peut être entièrement téléchargé lors d'un contact. Ensuite, nous avons généralisé un tel modèle pour tenir compte des téléchargements partiels, c'est-à-dire lors d'un contact un utilisateur peut télécharger une partie d'un contenu en fonction de la durée de contact. Bien que plus complexe d'un point de vue analytique, une description plus fine des téléchargements des contenus décrit mieux l'interaction entre les véhicules et les utilisateurs. Afin de valider nos résultats théoriques, nous avons effectué de nombreuses simulations basées sur des traces réelles. En particulier, nous avons soutenu qu'une répartition optimale du contenu entraîne une diminution de plus de la moitié de la charge totale de trafic dans des conditions réalistes.

Nous croyons que cette thèse peut être considérée comme un travail novateur parce qu'elle révèle le potentiel d'une infrastructure supplémentaire à bon marché. Une telle nouvelle infrastructure sera probablement disponible pour tout le monde dans un proche avenir. Également motivés par certaines premières tentatives de définition des normes, nous croyons certainement que les véhicules feront partie active de l'infrastructure cellulaire dans un futur proche. Notre étude peut être considérée comme pertinente car elle fournit des preuves de la potentialité du cloud de véhicules et peut accélérer son adoption.

Bibliography

3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2*. TS 36.300. 3rd Generation Partnership Project (3GPP), 2014. URL: http://www.3gpp.org/ftp/Specs/html-info/36300.htm.

Abrahamsson, Henrik and Mattias Nordmark. "Program Popularity and Viewer Behaviour in a Large TV-on-demand System". In: *Proceedings of the 2012 Internet Measurement Conference*. IMC '12. Boston, Massachusetts, USA: ACM, 2012, pp. 199–210. ISBN: 978-1-4503-1705-4. DOI: 10.1145/2398776.2398798. URL: http://doi.acm.org/10.1145/2398776.2398798.

Ahlehagh, H. and S. Dey. "Video-Aware Scheduling and Caching in the Radio Access Network". In: *IEEE/ACM Transactions on Networking* 22.5 (2014), pp. 1444–1462. ISSN: 1063-6692. DOI: 10.1109/TNET.2013.2294111.

Alliance, NGMN. *NGMN 5G White Paper*. https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf. 2015.

Analysis Mason. "M2M device connections and revenue: worldwide forecast 2014-2024". In: Analysis Mason Report, 2014.

Andrews, J. G., S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang. "What Will 5G Be?" In: *IEEE Journal on Selected Areas in Communications* 32.6 (2014), pp. 1065–1082. ISSN: 0733-8716.

Ao, Weng Chon and Konstantinos Psounis. "Distributed Caching and Small Cell Cooperation for Fast Content Delivery". In: *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc '15. Hangzhou, China: ACM, 2015, pp. 127–136. ISBN: 978-1-4503-3489-1. DOI: 10.1145/2746285.2746300. URL: http://doi.acm.org/10.1145/2746285.2746300.

Balasubramanian, Aruna, Ratul Mahajan, and Arun Venkataramani. "Augmenting Mobile 3G Using WiFi". In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*. San Francisco, California, USA: ACM, 2010, pp. 209–222. ISBN: 978-1-60558-985-5. DOI: 10.1145/1814433.1814456. URL: http://doi.acm.org/10.1145/1814433.1814456.

Bao, X., Y. Lin, U. Lee, I. Rimac, and R. R. Choudhury. "DataSpotting: Exploiting naturally clustered mobile devices to offload cellular traffic". In: *Proceedings IEEE INFOCOM*. 2013, pp. 420–424. DOI: 10.1109/INFCOM.2013.6566807.

Bastug, E., M. Bennis, and M. Debbah. "Living on the edge: The role of proactive caching in 5G wireless networks". In: *IEEE Communications Magazine* 52.8 (2014), pp. 82–89. ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6871674.

Baştuğ, E., M. Bennis, and M. Debbah. "Cache-enabled small cell networks: Modeling and tradeoffs". In: *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*. 2014, pp. 649–653. DOI: 10.1109/ISWCS.2014.6933434.

Baştuğ, E., J. L. Guénégo, and M. Debbah. "Proactive small cell networks". In: *ICT 2013*. 2013, pp. 1–5. DOI: 10.1109/ICTEL.2013.6632164.

Blasco, P. and D. Gündüz. "Learning-based optimization of cache content in a small cell base station". In: *2014 IEEE International Conference on Communications (ICC)*. 2014, pp. 1897–1903. DOI: 10.1109/ICC.2014.6883600.

BMW Vehicular CrowdCell. https://www.press.bmwgroup.com/global/article/detail/T0256442EN/bmw-presents-the-vehicular-crowdcell-at-the-mobile-world-congress-2016-in-barcelona-mobile-femtocells-help-to-optimize-future-mobile-radio-networks.

Borst, S., V. Gupta, and A. Walid. "Distributed Caching Algorithms for Content Distribution Networks". In: *INFOCOM, 2010 Proceedings IEEE*. 2010, pp. 1–9. DOI: 10.1109/INFCOM.2010.5461964.

Boyd, Stephen and Lieven Vandenberghe. *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004. ISBN: 0521833787.

Brunnström, Kjell et al. *Qualinet White Paper on Definitions of Quality of Experience*. Qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting, Novi Sad, March 12, 2013. Mar. 2013. URL: https://hal.archives-ouvertes.fr/hal-00977812.

Bychkovsky, Vladimir, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. "A Measurement Study of Vehicular Internet Access Using in Situ Wi-Fi Networks". In: *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*. MobiCom '06. Los Angeles, CA, USA: ACM, 2006, pp. 50–61. ISBN: 1-59593-286-0. DOI: 10.1145/1161089.1161097. URL: http://doi.acm.org/10.1145/1161089.1161097.

Cai, H., I. Koprulu, and N. B. Shroff. "Exploiting double opportunities for deadline based content propagation in wireless networks". In: *Proceedings IEEE INFOCOM*. 2013, pp. 764–772. DOI: 10.1109/INFCOM.2013.6566863.

Cha, Meeyoung, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. "Analyzing the Video Popularity Characteristics of Large-scale User Generated Content Systems". In: *IEEE/ACM Transactions on Networking (TON)* 17.5 (Oct. 2009), pp. 1357–1370. ISSN: 1063-6692. DOI: 10.1109/TNET.2008.2011358. URL: http://dx.doi.org/10.1109/TNET.2008.2011358.

Cheng, Nan, Ning Lu, Ning Zhang, Xuemin (Sherman) Shen, and Jon W. Mark. "Vehicular WiFi Offloading". In: *Veh. Commun.* 1.1 (Jan. 2014), pp. 13–21. ISSN: 2214-2096. DOI: 10.1016/j.vehcom.2013.11.002. URL: http://dx.doi.org/10.1016/j.vehcom.2013.11.002.

Choi, P., J. Gao, N. Ramanathan, M. Mao, S. Xu, C. C. Boon, S. A. Fahmy, and L. S. Peh. "A case for leveraging 802.11p for direct phone-to-phone communications". In: *2014 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. 2014, pp. 207–212. DOI: 10.1145/2627369.2627644.

Cisco. "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update". In: 2016-2021.

Coffman, Edward Grady and Peter J Denning. *Operating systems theory*. Vol. 973. Prentice-Hall Englewood Cliffs, NJ, 1973.

Cohen, E. and S. Shenker. "Replication Strategies in Unstructured Peer-to-peer Networks". In: *ACM SIGCOMM*. 2002. ISBN: 1-58113-570-X. DOI: 10.1145/633025.633043. URL: http://doi.acm.org/10.1145/633025.633043.

Conan, Vania, Jérémie Leguay, and Timur Friedman. "Characterizing Pairwise Intercontact Patterns in Delay Tolerant Networks". In: *Proceedings of the 1st International Conference on Autonomic Computing and Communication Systems (Autonomics)*. Rome, Italy: ICST, 2007, 19:1–19:9. ISBN: 978-963-9799-09-7. URL: http://dl.acm.org/citation.cfm?id=1365562.1365588.

Crane, Riley and Didier Sornette. "Viral, Quality, and Junk Videos on YouTube: Separating Content from Noise in an Information-Rich Environment". In: *AAAI Spring Symposium: Social Information Processing*. AAAI, 2008, pp. 18–20. URL: http://dblp.uni-trier.de/db/conf/aaaiss/aaaiss2008-6.html#CraneS08.

Dandapat, Sourav Kumar, Swadhin Pradhan, Niloy Ganguly, and Romit Roy Choudhury. "Sprinkler: Distributed Content Storage for Just-in-time Streaming". In: *Proceeding of the Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet)*. Taipei, Taiwan: ACM, 2013, pp. 19–24. ISBN: 978-1-4503-2074-0. DOI: 10.1145/2482985.2482986. URL: http://doi.acm.org/10.1145/2482985.2482986.

Dantzig, George B. “Discrete-Variable Extremum Problems”. In: *Operations Research* 5.2 (1957), pp. 266–277. ISSN: 0030364X, 15265463. URL: http://www.jstor.org/stable/167356.

Dehghan, Mostafa, Laurent Massoulié, Don Towsley, Daniel Sadoc Menasche, and Yong Chiang Tay. “A utility optimization approach to network cache design”. In: *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. San Francisco, United States, Apr. 2016. DOI: 10.1109/INFOCOM.2016.7524445. URL: https://hal.archives-ouvertes.fr/hal-01377841.

Dezsö, Z., E. Almaas, A. Lukács, B. Rácz, I. Szakadát, and A.-L. Barabási. “Dynamics of information access on the web”. In: *Phys. Rev. E* 73 (6 2006), p. 066132. DOI: 10.1103/PhysRevE.73.066132. URL: https://link.aps.org/doi/10.1103/PhysRevE.73.066132.

Erman, J., A. Gerber, K. K. Ramadrishnan, S. Sen, and O. Spatscheck. “Over the Top Video: The Gorilla in Cellular Networks”. In: *SIGCOMM, Proc. ACM*. 2011, pp. 127–136. ISBN: 978-1-4503-1013-0. DOI: 10.1145/2068816.2068829. URL: http://doi.acm.org/10.1145/2068816.2068829.

Everett III, Hugh. “Generalized Lagrange multiplier method for solving problems of optimum allocation of resources”. In: *Operations research* 11.3 (1963), pp. 399–417.

Floudas, C.A. and V. Visweswaran. “A global optimization algorithm (GOP) for certain classes of nonconvex NLPs”. In: *Computers & Chemical Engineering* 14.12 (1990), pp. 1397 –1417. ISSN: 0098-1354.

Forum, Small Cell. “Backhaul Technologies for Small Cells: Use Cases, Requirements and Solutions”. In: (2013).

Gao, G., 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)*. 2016, pp. 1–9. DOI: 10.1109/SAHCN.2016.7732980.

Garetto, M., E. Leonardi, and S. Traverso. “Efficient analysis of caching strategies under dynamic content popularity”. In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2015, pp. 2263–2271. DOI: 10.1109/INFOCOM.2015.7218613.

Gill, Phillipa, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. “Youtube Traffic Characterization: A View from the Edge”. In: *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*. IMC ’07. San Diego, California, USA: ACM, 2007, pp. 15–28. ISBN: 978-1-59593-908-1. DOI: 10.1145/1298306.1298310. URL: http://doi.acm.org/10.1145/1298306.1298310.

Golrezaei, N., P. Mansourifard, A. F. Molisch, and A. G. Dimakis. "Base-Station Assisted Device-to-Device Communications for High-Throughput Wireless Video Networks". In: *IEEE Transactions on Wireless Communications* 13.7 (2014), pp. 3665–3676. ISSN: 1536-1276. DOI: 10.1109/TWC.2014.2316817.

Golrezaei, N., K. Shanmugam, AG. Dimakis, AF. Molisch, and G. Caire. "Femto-Caching: Wireless video content delivery through distributed caching helpers". In: *IEEE INFOCOM*. 2012. DOI: 10.1109/INFCOM.2012.6195469.

Golrezaei, Negin, Alexandros G. Dimakis, and Andreas F. Molisch. "Wireless Device-to-Device Communications with Distributed Caching". In: *CoRR* abs/1205.7044 (2012). URL: http://arxiv.org/abs/1205.7044.

Golrezaei, Negin, Andreas F Molisch, Alexandros G Dimakis, and Giuseppe Caire. "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution". In: *IEEE Communications Magazine* 51.4 (2013), pp. 142–149.

Gorski, Jochen, Frank Pfeuffer, and Kathrin Klamroth. "Biconvex sets and optimization with biconvex functions: a survey and extensions". In: *Mathematical Methods of Operations Research* 66.3 (2007), pp. 373–407. ISSN: 1432-5217. DOI: 10.1007/s00186-007-0161-1. URL: http://dx.doi.org/10.1007/s00186-007-0161-1.

Green, Jeremy. *The connected car market will grow rapidly to be worth USD 284B by 2024*. https://machinaresearch.com/report/the-connected-car-market-will-grow-rapidly-to-be-worth-usd284-billion-by-2024/. 2014.

Guan, Y., Y. Xiao, H. Feng, C.-C. Shen, and L. J. Cimini Jr. "MobiCacher: Mobility-Aware Content Caching in Small-Cell Networks". In: *CoRR* abs/1407.1307 (2014). URL: http://arxiv.org/abs/1407.1307.

Gupta, P. and P. R. Kumar. "The capacity of wireless networks". In: *IEEE Transactions on Information Theory* 46.2 (2000), pp. 388–404. ISSN: 0018-9448. DOI: 10.1109/18.825799.

Ha, S., S. Sen, C. Joe-Wong, Y. Im, and M. Chiang. "TUBE: Time-dependent Pricing for Mobile Data". In: *ACM SIGCOMM*. 2012. ISBN: 978-1-4503-1419-0. DOI: 10.1145/2342356.2342402. URL: http://doi.acm.org/10.1145/2342356.2342402.

Hadaller, David, Srinivasan Keshav, Tim Brecht, and Shubham Agarwal. "Vehicular Opportunistic Communication Under the Microscope". In: *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*. MobiSys '07. San Juan, Puerto Rico: ACM, 2007, pp. 206–219. ISBN: 978-1-59593-614-1. DOI: 10.

1145/1247660.1247685. URL: http://doi.acm.org/10.1145/1247660.1247685.

Hall, Peter. "Heavy traffic approximations for busy period in an M/G/∞ queue". In: *Stochastic Processes and their Applications* 19.2 (1985), pp. 259 –269. ISSN: 0304-4149. DOI: http://dx.doi.org/10.1016/0304-4149(85)90028-6. URL: http://www.sciencedirect.com/science/article/pii/0304414985900286.

Hameed Mir, Zeeshan and Fethi Filali. "LTE and IEEE 802.11p for vehicular networking: a performance evaluation". In: *EURASIP Journal on Wireless Communications and Networking* 2014.1 (2014), pp. 1–15. DOI: 10.1186/1687-1499-2014-89. URL: http://dx.doi.org/10.1186/1687-1499-2014-89.

Han, B., P. Hui, V. S. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan. "Mobile Data Offloading through Opportunistic Communications and Social Participation". In: *IEEE Transactions on Mobile Computing* 11.5 (2012), pp. 821–834. ISSN: 1536-1233. DOI: 10.1109/TMC.2011.101.

Han, W., A. Liu, and V. K. N. Lau. "Tradeoff between PHY caching and core network caching in cellular networks". In: *2016 IEEE International Conference on Communications (ICC)*. 2016, pp. 1–6. DOI: 10.1109/ICC.2016.7511521.

Harchol-Balter, Mor. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. 1st. New York, NY, USA: Cambridge University Press, 2013. ISBN: 1107027500, 9781107027503.

Hossfeld, T., S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen. "Initial delay vs. interruptions: Between the devil and the deep blue sea". In: *2012 Fourth International Workshop on Quality of Multimedia Experience*. 2012, pp. 1–6. DOI: 10.1109/QoMEX.2012.6263849.

Hyytiä, E., J. Virtamo, P. Lassila, J. Kangasharju, and J. Ott. "When does content float? Characterizing availability of anchored information in opportunistic content sharing". In: *2011 Proceedings IEEE INFOCOM*. 2011, pp. 3137–3145. DOI: 10.1109/INFCOM.2011.5935160.

Ji, M., G. Caire, and A. F. Molisch. "Wireless Device-to-Device Caching Networks: Basic Principles and System Performance". In: *IEEE Journal on Selected Areas in Communications* 34.1 (2016), pp. 176–189. ISSN: 0733-8716. DOI: 10.1109/JSAC.2015.2452672.

Kannan, Ravindran and Clyde L Monma. "On the computational complexity of integer programming problems". In: *Optimization and Operations Research*. Springer, 1978, pp. 161–172.

Karagiannis, T., J. Y. Le Boudec, and M. Vojnovic. "Power Law and Exponential Decay of Intercontact Times between Mobile Devices". In: *IEEE Transactions on Mobile Computing* 9.10 (2010), pp. 1377–1390. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.99.

Karagiannis, Thomas, Jean-Yves Le Boudec, and Milan Vojnović. "Power Law and Exponential Decay of Inter Contact Times Between Mobile Devices". In: *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*. MobiCom '07. Montr&#233;al, Qu&#233;bec, Canada: ACM, 2007, pp. 183–194. ISBN: 978-1-59593-681-3. DOI: 10.1145/1287853.1287875. URL: http://doi.acm.org/10.1145/1287853.1287875.

Karlin, S. and H.E. Taylor. *A First Course in Stochastic Processes*. Elsevier Science, 2012. ISBN: 9780080570419. URL: https://books.google.fr/books?id=dSDxjX9nmmMC.

Kim, Ho-Yeon, Dong-Min Kang, Jun-Ho Lee, and Tai-Myoung Chung. "A Performance Evaluation of Cellular Network Suitability for VANET". In: *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 6.4 (2012), pp. 448 –451. ISSN: PISSN:2010-376X, EISSN:2010-3778. URL: http://waset.org/Publications?p=64.

Lee, J. G., S. Moon, and K. Salamatian. "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. 2010, pp. 623–630. DOI: 10.1109/WI-IAT.2010.209.

Lee, K., J. Lee, Y. Yi, I. Rhee, and S. Chong. "Mobile Data Offloading: How Much Can WiFi Deliver?" In: *IEEE/ACM Transactions on Networking* 21.2 (2013), pp. 536–550. ISSN: 1063-6692. DOI: 10.1109/TNET.2012.2218122.

Lee, K., S. Hong, S. J. Kim, I. Rhee, and S. Chong. "SLAW: A New Mobility Model for Human Walks". In: *IEEE INFOCOM*. 2009, pp. 855–863. DOI: 10.1109/INFCOM.2009.5061995.

Li, Yong, Guolong Su, Pan Hui, Depeng Jin, Li Su, and Lieguang Zeng. "Multiple Mobile Data Offloading Through Delay Tolerant Networks". In: *Proceedings of the 6th ACM Workshop on Challenged Networks*. CHANTS '11. Las Vegas, Nevada, USA: ACM, 2011, pp. 43–48. ISBN: 978-1-4503-0870-0. DOI: 10.1145/2030652.2030665. URL: http://doi.acm.org/10.1145/2030652.2030665.

Lin, Wei-Yen, Mei-Wen Li, Kun-Chan Lan, and Chung-Hsien Hsu. "Quality, Reliability, Security and Robustness in Heterogeneous Networks". In: ed. by Xi Zhang and Daji Qiao. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 559–570. ISBN: 978-3-642-29222-4. DOI: 10.1007/978-3-642-29222-4_39. URL: http://dx.doi.org/10.1007/978-3-642-29222-4_39.

Liu, A. and V. K. N. Lau. “Cache-Enabled Opportunistic Cooperative MIMO for Video Streaming in Wireless Systems”. In: *IEEE Transactions on Signal Processing* 62.2 (2014), pp. 390–402. ISSN: 1053-587X. DOI: 10.1109/TSP.2013.2291211.

– “Exploiting Base Station Caching in MIMO Cellular Networks: Opportunistic Cooperation for Video Streaming”. In: *IEEE Transactions on Signal Processing* 63.1 (2015), pp. 57–69. ISSN: 1053-587X. DOI: 10.1109/TSP.2014.2367473.

Lu, N., N. Cheng, N. Zhang, X. Shen, and J. W. Mark. “Connected Vehicles: Solutions and Challenges”. In: *IEEE Internet of Things Journal* 1.4 (2014), pp. 289–299. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2327587.

Maddah-Ali, Mohammad Ali and Urs Niesen. “Decentralized Coded Caching Attains Order-optimal Memory-rate Tradeoff”. In: *IEEE/ACM Trans. Netw.* 23.4 (Aug. 2015), pp. 1029–1040. ISSN: 1063-6692. DOI: 10.1109/TNET.2014.2317316. URL: http://dx.doi.org/10.1109/TNET.2014.2317316.

Mahmood, A., C. Casetti, C. F. Chiasserini, P. Giaccone, and J. Harri. “Mobility-aware edge caching for connected cars”. In: *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2016, pp. 1–8.

Mamun, Md Ali Al, Khairul Anam, and Md Fakhrul Alam. “Deployment of Cloud Computing into VANET to Create Ad Hoc Cloud Network Architecture”. In: 2012.

Martello, Silvano and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990. ISBN: 0-471-92420-2.

Mehmeti, F. and T. Spyropoulos. “Is it worth to be patient? Analysis and optimization of delayed mobile data offloading”. In: *IEEE INFOCOM Conference on Computer Communications*. 2014, pp. 2364–2372. DOI: 10.1109/INFOCOM.2014.6848181.

Naylor, David, Kyle Schomp, Matteo Varvello, Ilias Leontiadis, et al. “Multi-Context TLS (mcTLS): Enabling Secure In-Network Functionality in TLS”. In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. SIGCOMM ’15. London, United Kingdom: ACM, 2015, pp. 199–212. ISBN: 978-1-4503-3542-3. DOI: 10.1145/2785956.2787482. URL: http://doi.acm.org/10.1145/2785956.2787482.

Nikolikj, Vladimir and Toni Janevski. “A Cost Modeling of High-capacity LTE-advanced and IEEE 802.11ac based Heterogeneous Networks, Deployed in the 700 MHz, 2.6 GHz and 5 GHz Bands”. In: *Procedia Computer Science* 40 (2014). Fourth International Conference on Selected Topics in Mobile & Wireless Networking (MoWNet’2014), pp. 49 –56. ISSN: 1877-0509. DOI: http://dx.doi.org/10.1016/j.procs.2014.10.030. URL: http://www.sciencedirect.com/science/article/pii/S1877050914013970.

Ostovari, P., J. Wu, and A. Khreishah. “Efficient Online Collaborative Caching in Cellular Networks with Multiple Base Stations”. In: *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2016, pp. 136–144. DOI: 10.1109/MASS.2016.027.

Ott, J. and D. Kutscher. “Drive-thru Internet: IEEE 802.11b for "automobile" users”. In: *IEEE INFOCOM 2004*. Vol. 1. 2004, p. 373. DOI: 10.1109/INFCOM.2004.1354509.

Paschos, G. S., S. Gitzenis, and L. Tassiulas. “The effect of caching in sustainability of large wireless networks”. In: *10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. 2012, pp. 355–360.

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

Poularakis, K., G. Iosifidis, A. Argyriou, and L. Tassiulas. “Video delivery over heterogeneous cellular networks: Optimizing cost and performance”. In: *IEEE INFOCOM Conference on Computer Communications*. 2014, pp. 1078–1086. DOI: 10.1109/INFOCOM.2014.6848038.

Raghavan, Prabhakar and Clark D. Tompson. “Randomized rounding: A technique for provably good algorithms and algorithmic proofs”. In: *Combinatorica* 7.4 (1987), pp. 365–374. ISSN: 1439-6912. DOI: 10.1007/BF02579324. URL: http://dx.doi.org/10.1007/BF02579324.

Rappaport, T. S. et al. “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!” In: *IEEE Access* 1 (2013), pp. 335–349. ISSN: 2169-3536.

Robson, J. *Small Cell Deployment Strategies and Best Practice Backhaul*. 2012.

Ross, S.M. *Stochastic Processes*. Wiley series in mathematical statistics. Probability and mathematical statistics. Wiley, 1983. ISBN: 9780471099420. URL: https://books.google.fr/books?id=Hj7bAAAAMAAJ.

Sapountzis, N., S. Sarantidis, T. Spyropoulos, N. Nikaein, and U. Salim. “Reducing the energy consumption of small cell networks subject to QoE constraints”. In: *2014 IEEE Global Communications Conference*. 2014, pp. 2485–2491. DOI: 10.1109/GLOCOM.2014.7037181.

Sapountzis, Nikolaos, Thrasyvoulos Spyropoulos, Navid Nikaein, and Umer Salim. “Optimal downlink and uplink user association in Backhaul-limited HetNets”. In: *INFOCOM 2016, IEEE International Conference on Computer Communications, 10-15 April 2016, San Francisco, CA, USA*. San Fransisco, UNITED STATES, Apr. 2016. URL: http://www.eurecom.fr/publication/4814.

Schmidli, H. *Lecture Notes on Risk Theory*.

Senza Fili Consulting. "The economics of small cells and Wi-Fi offload". In: (2013).

Sermpezis, Pavlos and Thrasyvoulos Spyropoulos. "Not All Content is Created Equal: Effect of Popularity and Availability for Content-centric Opportunistic Networking". In: *Proceedings of the 15th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Philadelphia, Pennsylvania, USA: ACM, 2014, pp. 103–112. ISBN: 978-1-4503-2620-9. DOI: 10.1145/2632951.2632976. URL: http://doi.acm.org/10.1145/2632951.2632976.

Sesia, Stefania, Issam Toufik, and Matthew Baker. *LTE, The UMTS Long Term Evolution: From Theory to Practice*. Wiley Publishing, 2009. ISBN: 0470697164, 9780470697160.

Shanmugam, K., N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire. "Femto-Caching: Wireless Content Delivery Through Distributed Caching Helpers". In: *IEEE Transactions on Information Theory* 59.12 (2013), pp. 8402–8413. ISSN: 0018-9448. DOI: 10.1109/TIT.2013.2281606.

Smartphone on Wheels. http://www.economist.com/news/technology-quarterly/21615060-way-cars-are-made-bought-and-driven-changing-mobile-communications.

Szabo, Gabor and Bernardo A. Huberman. "Predicting the Popularity of Online Content". In: *Communications of the ACM* 53.8 (Aug. 2010), pp. 80–88. ISSN: 0001-0782. DOI: 10.1145/1787234.1787254. URL: http://doi.acm.org/10.1145/1787234.1787254.

Tasiopoulos, Argyrios G., Ioannis Psaras, and George Pavlou. "Mind the Gap: Modelling Video Delivery Under Expected Periods of Disconnection". In: *Proceedings of the 9th ACM MobiCom Workshop on Challenged Networks*. CHANTS '14. Maui, Hawaii, USA: ACM, 2014, pp. 13–18. ISBN: 978-1-4503-3071-8. DOI: 10.1145/2645672.2645680. URL: http://doi.acm.org/10.1145/2645672.2645680.

Thompson, Nathanael, Riccardo Crepaldi, and Robin Kravets. "Locus: A Location-based Data Overlay for Disruption-tolerant Networks". In: *Proceedings of the 5th ACM Workshop on Challenged Networks*. CHANTS '10. Chicago, Illinois, USA: ACM, 2010, pp. 47–54. ISBN: 978-1-4503-0139-8. DOI: 10.1145/1859934.1859945. URL: http://doi.acm.org/10.1145/1859934.1859945.

Traverso, Stefano, Mohamed Ahmed, Michele Garetto, Paolo Giaccone, Emilio Leonardi, and Saverio Niccolini. "Temporal Locality in Today's Content Caching: Why It Matters and How to Model It". In: *SIGCOMM Comput. Commun. Rev.* 43.5 (Nov. 2013), pp. 5–12. ISSN: 0146-4833. DOI: 10.1145/2541468.2541470. URL: http://doi.acm.org/10.1145/2541468.2541470.

Trifunovic, Sacha, Andreea Picu, Theus Hossmann, and Karin Anna Hummel. "Slicing the Battery Pie: Fair and Efficient Energy Usage in Device-to-device Communication via Role Switching". In: *Proceedings of the 8th ACM MobiCom Workshop on Challenged Networks*. CHANTS '13. Miami, Florida, USA: ACM, 2013, pp. 31–36. ISBN: 978-1-4503-2363-5. DOI: 10.1145/2505494.2505496. URL: http://doi.acm.org/10.1145/2505494.2505496.

Veniam. https://veniam.com/.

Vigneri, Luigi, Thrasyvoulos Spyropoulos, and Chadi Barakat. "Low Cost Video Streaming through Mobile Edge Caching: Modeling and Optimization". In: *IEEE Transactions on Mobile Computing* (2017).

– "Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud". In: MSWiM (2017).

– "Quality of Experience-Aware Mobile Edge Caching through a Vehicular Cloud". In: *IEEE Transactions on Mobile Computing* (2017).

– "Storage on wheels: Offloading popular contents through a vehicular cloud". In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2016, pp. 1–9. DOI: 10.1109/WoWMoM.2016.7523506.

– "Streaming Content from a Vehicular Cloud". In: *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. CHANTS '16. New York City, New York: ACM, 2016, pp. 39–44. ISBN: 978-1-4503-4256-8. DOI: 10.1145/2979683.2979684. URL: http://doi.acm.org/10.1145/2979683.2979684.

Wang, X., M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung. "Cache in the air: exploiting content caching and delivery techniques for 5G systems". In: *IEEE Communications Magazine* 52.2 (2014), pp. 131–139. ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6736753.

Wang, X., M. Chen, Z. Han, D. O. Wu, and T. T. Kwon. "TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks". In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 2014, pp. 2346–2354. DOI: 10.1109/INFOCOM.2014.6848179.

Wendell, Richard E. and Arthur P. Hurter. "Minimization of a Non-Separable Objective Function Subject to Disjoint Constraints". In: *Operations Research* 24.4 (Aug. 1976), pp. 643–657. ISSN: 0030-364X. DOI: 10.1287/opre.24.4.643. URL: http://dx.doi.org/10.1287/opre.24.4.643.

Whitbeck, John, Yoann Lopez, JéRéMie Leguay, Vania Conan, and Marcelo Dias De Amorim. "Fast Track Article: Push-and-track: Saving Infrastructure Bandwidth Through Opportunistic Forwarding". In: *Pervasive Mob. Comput.* 8.5 (Oct. 2012), pp. 682–697. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2012.02.001. URL: http://dx.doi.org/10.1016/j.pmcj.2012.02.001.

Woo, Shinae, Eunyoung Jeong, Shinjo Park, Jongmin Lee, Sunghwan Ihm, and KyoungSoo Park. "Comparison of Caching Strategies in Modern Cellular Backhaul Networks". In: *ACM MobiSys*. 2013.

Zeni, Mattia, Daniele Miorandi, and Francesco De Pellegrini. "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, 2013.

Zhang, F., Chenren Xu, Y. Zhang, K. K. Ramakrishnan, S. Mukherjee, R. Yates, and Thu Nguyen. "EdgeBuffer: Caching and prefetching content at the edge in the MobilityFirst future Internet architecture". In: *IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2015, pp. 1–9. DOI: 10.1109/WoWMoM.2015.7158137.

Zhang, Y., J. Zhao, and G. Cao. "Roadcast: A Popularity Aware Content Sharing Scheme in VANETs". In: *29th IEEE International Conference on Distributed Computing Systems*. 2009, pp. 223–230. DOI: 10.1109/ICDCS.2009.19.

Zhao, J. and G. Cao. "VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks". In: *IEEE Transactions on Vehicular Technology* 57.3 (2008), pp. 1910–1922. ISSN: 0018-9545. DOI: 10.1109/TVT.2007.901869.

Zheng, K., Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou. "Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions". In: *IEEE Communications Surveys Tutorials* 17.4 (2015), pp. 2377–2396. ISSN: 1553-877X. DOI: 10.1109/COMST.2015.2440103.