Information Delivery: Past Struggles & New Directions

Prométhée Spathis

UPMC
SORBONNE UNIVERSITÉS

Habilitation to Supervise Research Thesis

Information Delivery
Past Struggles & New Directions

Prométhée **Spathis**

Submitted in total fulfilment of the requirements
of the degree of Habilitation à Diriger des Recherches

Defended on December 20, 2016

Reviewers:	Ellen Zegura	Prof. at Georgia Institute of Technology
	Jim Kurose	Prof. at University of Massachusetts Amherst
	Olivier Festor	Prof. at TELECOM Nancy
Examiners:	Vania Conan	Senior Researcher at Thalès Communications & Security
	Parick Senac	Head of Research at ENAC
	Serge Fdida	Prof. at Université Pierre et Marie Curie

Contents

Part A

Introduction

Preface

This document presents a summary of the research I have been conducting as an assistant professor at Université Pierre et Marie Curie. The main topic of my research is the design and the analysis of information delivery systems in the context of challenging environments and disruptive usage patterns. The main driving motivation of much of my activities can be stated as follows: *how to handle the mobility in the many incarnations it can take, whether we consider the migration of content between fixed repositories or the movements of the hosting nodes if mobile?* In my research, I address the mobility, first as a challenge and then, as an enabler with regard to information delivery.

The work I present in this document results of collaborations conducted with the following Ph.D. students I previously co-advised: Yosra Barouni (ISIMG), Mohamed Diallo (AITEK), Raul Adrian Gorcitz (Inria), Benjamin Baron (UPMC), and also Salah-Eddine Belouanas (UPMC) who I am currently co-advising. Many sentences and paragraphs of this document were taken from papers jointly written with the aforementioned Ph.D. students. This document would not be possible without their contribution and the significant support from the main supervisors Professor Serge Fdida (UPMC) and Research Director Marcelo Dias de Amorim (CNRS/UPMC).

Several parts of my work have been applied and implemented in the framework of the following research projects: the EU funded projects ANA (Autonomous network architecture), FP6 IP project (01/06–12/09), ACCA (Autonomic communication: Coordination action), FP6 Open FET Coordination Action (12/04–06/06), FP7 MOTO (Mobile opportunistic traffic offloading) (11/12–10/15), the STIC-Amsud (French scientific-technological cooperation programme with South America) projects SCAN (Self-conscious ambient networks) (01/09–12/10), CUDEN (Collaborative user driven networking) (01/12–12/13), and MOSAIC (Mobile crowd sensing and data offloading in collaborative networks) (01/14–12/15), and the French national projects ANR CONNECT (12/10–11/12), RESCUE (12/10–11/13), DataTweet (10/13–11/16). These projects provided a fertile research environment for constructive discussions and fruitful collaborations with colleagues and partners.

While preparing this manuscript, I had to face two difficulties. The first was to provide a comprehensive overview of my research activities in 50 pages. The second was to render faithfully the rationale behind my approach to research, from the early stage of a 'hunch' to the dissemination of findings. I do hope both have been adequately addressed. In the following, I first give the rationale lying behind the contributions presented in this document. Figure A.1 gives a tree representation showing the evolution of the research topics I have been working on. I then give the outline of this document.

Contribution rationale

At the time I took on my first academic position, my background in research was, among others, focused on Active and Programmable Networks. A natural transition to carry on my research activities was to investigate the emerging areas of Autonomic Networking and Self-Managing Networks which were attracting a significant part of the research efforts undertaken in developing the future Internet.

At the same time, many interesting and progressive ideas were coming from the research community invested in defining content delivery mechanisms for the next generation of Internet. I decided to investigate the design of routing as one of the major function of any network infrastructure. The idea was to evaluate the benefits of the so-called self-$\star$ properties in reducing the cost and the complexity of managing the routing architectures.

The outcome of this work, not presented in this thesis for space reasons, came in the framework of a filtering- and content-based pub/sub system. Subscribers express their interests in receiving specific content by means of selected attribute values. These values are then used by en-route filtering and selective forwarding of the messages carrying a matching content, once released by the publishers.

At that same period, Content Centric Networking (CCN) was emerging as a promising paradigm. I decided to follow up on my work on content delivery by taking advantage of the momentum generated by the proposal of CCN. The results of this work are part of the three contributions I present in the next section.

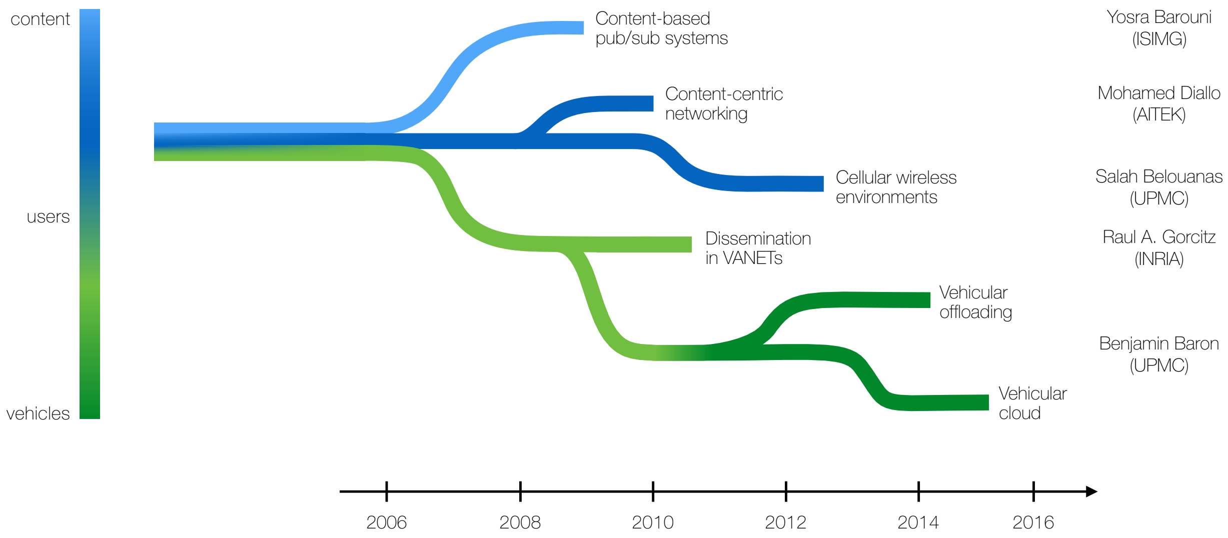


Figure A.1. Topic evolution tree

Content, the big 'C' of computer networks (Part B)

In this first part of the work I present in this thesis, I address a number of limitations that plague conventional data networks. First, I tackle the limitations of the Internet from the perspective of the principles that have driven its design. I then address the mismatch between the available capacity of cellular wireless networks and the increasing demand for content on the move. Finally, I investigate the problem of data dissemination in vehicular networks.

Leveraging content replication for data-intensive applications (Chapter B.1)
My first contribution to content delivery came into the form of a scalable inter-domain routing protocol designed to complement CCN existing communication framework. This protocol relies on an Interest-based approach for discovering copies of replicated content. The caching capabilities of content routers are used to store content objects but also previously learnt routes. The broad motivation of this work was to meet the needs of content related services and data-intensive applications.

Leveraging gatherings in cellular wireless networks (Chapter B.2)
My second contribution came into the context of cellular wireless networks. This work addresses the delivery of popular delay-tolerant content while coping with the resource scarcity of those networks. The main contribution is the design of a transmission-delaying strategy based on a predictive heuristic regarding the gathering ratio of mobile users interested in the same content within a cell.

The rest of the research work presented in this document finds its essence in the discussion initiated within the framework of a research grant funded in 2009 by Toyota InfoTechnology Center. The objective was to investigate what would be the use of vehicles in 10 years by then. The idea was to provide new market opportunities for car manufacturers industry by rethinking the role of the vehicle in the context of the future Internet.

Data dissemination in vehicular networks (Chapter B.3)
I conducted a preliminary study of the state of the art to kickoff my research activities in the domain of vehicular networks, unknown to me when I started this work. This study gave me the opportunity to create a new taxonomy of the technical advancement for disseminating data in vehicular environments. This taxonomy helped me gain a better understanding of the challenges raised by the mobility in the context of vehicular networks. I addressed some of these challenges by the proposal of a new dissemination protocol. Despite its efficiency, this protocol was one among many preceding and following.

The review of the relevant research literature related to vehicular networks also revealed that most studies considered application scenarios oriented toward the vehicle, the driver, or the passengers. According to those scenarios, the vehicles collaborate to offer safety and convenience to the drivers or entertainment to the passengers. The common objective of these studies was to provide the required technological advancement so vehicles can communicate while eliminating the need of infrastructure support. The vehicles are considered as the end-points of communications involving the delivery of content they produce or consume. At the same time, the vehicles may be involved as intermediate relays to overcome the intermittent connectivity resulting from the highly mobile environment where they operate.

Tapping into everyday mobility (Part C)

My first contributions to VANETs confirmed the need of tackling the field of vehicular networks from a new perspective. In the continuation of my work, I decided to adopt a radical new approach. In my vision, the existing mobility of vehicles can be regarded as an enabler to extend or replace an infrastructure network such as the Internet. Once equipped with storage capabilities, the flows of vehicles and the road they travel in their line of travel can be considered as a communication infrastructure on its own. The density of the vehicles as well as the scale and coverage of the roads they travel are all factors that advocate for the use of the road network as an alternative communication channel.

My first paper on this topic was published in January 2011, before the sharing economy was recognized as one of the ten ideas that will change the world[1]. Examples that fall under the same umbrella with my work include ridesharing and carsharing. With services such as Car2Go and ZipCar or car-hailing apps such as Uber and Lyft, car owners or their drivers, if different, can monetize mobility as a resource they can share on demand. Customers are most usually charged for the distance and duration of the trip or the time they drive. In my vision, mobility, if properly characterized, can be allocated and managed as a network resource.

Equipping mobile entities with storage capabilities may appear somewhat akin to the store, carry, and forward paradigm underlying DTNs (delay tolerant networks). While the focus of DTNs has been on exploiting the node encounters to pass data hop-by-hop asynchronously, my work exploits the benefits of the carry phase. To realize my vision, my research took two main complementary directions. My first objective was to translate the attributes of a road traffic into networking quantities. My second objective was to find means of mitigating the randomness of node mobility. My contributions to those two objectives were made in the context of various innovative applications including data offloading and cloud-like services.

Vehicular data offloading (Chapter C.1)

My first proposal came into the context of traffic offloading which refers to techniques proposed to alleviate the ever-growing traffic load that drags down the delivery performance of conventional data networks. I proposed a vehicular-based offloading system for enhancing the transport capacity of legacy data networks such as the Internet. This system leverages on the increasing number of conventional vehicles driven and miles traveled to opportunistically offload large chunks of data from the Internet. Vehicles are equipped with storage devices and act as data carriers whilst being driven for daily routine journeys.

The benefits of using conventional vehicles for offloading data traffic come from the capacity which results of the vehicles' data storage size and the distance they travel. Nevertheless, relying only on the vehicles making the trips all the way from the source to the destination is expected to limit these benefits. To improve the capacity of this offloading system, I proposed to deploy a collection of *offloading spots*. An offloading spot is a wireless data storage equipment acting as a data exchange relay point located where vehicles usually stop. Vehicles can drop off data along their line of travel for later pick-up by another vehicle.

To assess the concept of offloading data on vehicles, I proposed a first placement strategy in the context of a network of charging stations for electric vehicles (EVs). The charging stations are equipped to act as offloading spots where data is loaded on or off the EVs, without the drivers being aware, while charging their batteries as they usually do. The location of the charging stations is determined by solving a facility-allocation problem that minimizes the number of stations needed to satisfy all the charging demands of EV batteries. Once the offloading spots deployed, I tackled relevant challenges from the point of view of offloading by answering the following two questions: (*i*) How to compute the road path matching the performance requirements of a data transfer and (*ii*) how to configure the sequence of offloading spots involved in the transfer.

As a continuation to this work on traffic offloading, I proposed two additional placement strategies designed for two variants of the offloading spots in the context of two vehicular cloud services.

Vehicular cloud services (Chapter C.2)

A second placement strategy for the offloading spots came into the context of a vehicular cloud-like file sharing system. The offloading spots are now occupied by repositories strategically pre-positioned where files are replicated without relying on a conventional infrastructure-based network. The existing mobility of vehicles is opportunistically used to synchronize the files among all the repositories. I proposed a placement algorithm that finds the locations of a target number of repositories so as to satisfy a maximum number of user requests and so repositories can synchronize their content by the existing movements of mobile nodes.

A third placement strategy was proposed in the context of the migration of virtual machines hosted on vehicles. Instead of considering the offloading spots pre-positionned, I investigated the benefit of floating offloading spots in the context of a large scale vehicular network where the vehicle resources including storage, but also processing or sensing are virtualized. The purpose of this study was to determine whether the migration of virtual machines (of typically 200 MB) is feasible between vehicles in contact. The offloading spots are now dematerialized and refer to area where vehicles come

[1] http://ti.me/13XZQOI

in contact often and for long period of time compared to other areas. The objective of the placement strategy is here to select the locations that maximize the contact opportunities between vehicles.

Based on the various strategies developed in my work on vehicular data offloading and cloud-like services, I conclude this second part with a survey of the relevant literature on mobility-assisted transmission systems. I review the methods used to pass the data among the multiple relay entities involved in a given transfer.

Document outline

This document is organized in four parts, including this introduction (Part A) and a last part (Part D) that concludes this thesis with a summary and a number of issues as a plan for future research. The two middle parts (Part B and Part C) form the core of this document. Each part is divided into chapters where I present one or more selected publications listed inside square grey shaded boxes. I start each part with a brief introduction and then give a chapter-level summary listing the main contributions. In every chapters, the key takeways of each section are highlighted in pullout square boxes using a distinctive typeface. I present below a quick overview of each part:

- Part B (three chapters) presents three pieces of work related with content delivery in the context of various environments. In Chapter B.1, I consider the core of large-scale conventional networks such as the Internet. I address the limitations resulting from the host-centric design of the Internet. The contribution of this chapter is a query-based search approach that leverages the replication of popular content. In Chapter B.2, I address the mismatch between the available capacity of cellular wireless networks and the increasing demand for content on the move. I propose a radio resource-saving strategy for disseminating delay-tolerant content in cellular environments. In Chapter B.3, I consider the problem of disseminating content in vehicular networks.
- Part C (two chapters) presents a number of contributions taking advantage of the alternative communication channel resulting from the movements of entities such as vehicles equipped with storage capabilities. I use this alternative communication channel to supplement or in replacement of conventional infrastructure-based data networks. I exploit the existing mobility of surrounding entities to overcome various limitations of conventional data networks. In Chapter C.1, I use the routine journeys daily taken by private cars equipped with data storage devices to extend cost-effectively the capacity of the Internet. In Chapter C.2, I present two vehicular cloud services exploiting the mobility of public transit buses in urban scenarios.

Part B
Content, the Big 'C' of Computer Networking

In this first part of my work, my research addresses a number of limitations that plague conventional data networks. The term data network is used in their commonly well-understood meaning and refer to large infrastructure-based networks including the Internet and then cellular wireless networks. First, I tackle the limitations of the Internet from the perspective of the principles that have driven its design. I then address the mismatch between the available capacity of cellular wireless networks and the increasing demand for content on the move.

In Chapter B.1, I address the mismatch resulting from the so-called host-centric principle with regard to the needs of content related services. I exploit the ability to access content in a location-independent fashion introduced by the paradigm of content centric networking. Content refers to the data itself and the aim is to discover the copies of popular content in a large-scale infrastructure-based network. This contribution came after a preliminary work into the context of the content-based pub/sub systems.

In Chapter B.2, I then address the network capacity issue mobile operators are facing due to the exponential growth of mobile traffic expected for the upcoming years. In this work, in addition to the relationships between users and content, I take into account the growing demand of users for data while on the move. I exploit their tendency in gathering together to deliver delay-tolerant content.

After addressing some of the limitations of traditional data networks, I investigate the problem of data dissemination in vehicular networks. Chapter B.3 first presents a comparative analysis of the most relevant existing broadcast-based data dissemination schemes. I then describe a low overhead and disconnection tolerant data dissemination protocol that mitigates broadcast storms without relying on road map information.

The contributions presented in this first part are organized as follows:

Content-based publish/subscribe

> (Publication [1]: Y. Barouni, P. Spathis, S. Fdida, "Content Centric Routing for the Autonomic Networks", *2nd International Conference on Autonomic Computing and Communication Systems (Autonomics)*, 2008.
> (Publication [2]: Y. Barouni, P. Spathis, S. Fdida, "Algorithme distribué de filtrage et de routage orienté contenu", *10ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, 2008.
> (Publication [3]: Y. Barouni, P. Spathis, S. Fdida, "Magnet: A Content Centric Routing Protocol for Large Scale Networks", *4th International Conference on Future Internet Technologies (CFI)*, 2009.
> (Publication [4]: P. Spathis, M. Bicoudo, "ANA: Autonomic Network Architecture", *Autonomic Network Management Principles: From Concepts to Applications*, Academic Press, 2011.
> (Publication [5]: M. Diallo, S. Fdida, P. Spathis, "Towards Extreme Scale Content-Based Networking for the Next Generation Internet", *IEICE Transactions on Communications*, 2011.
> (Ph.D. thesis of Yosra Barouni, UPMC Sorbonne Universités, 2011. Co-supervised by Serge Fdida.)
> (Ph.D. thesis of Mohamed Diallo, UPMC Sorbonne Universités, 2013. Co-supervised by Serge Fdida.)

My first contributions to the area of content-based networking came into the framework of publish/subscribe systems. Those contributions are not presented in this document for space reasons. In these systems, subscribers express their interests in receiving specific content by means of selected attribute values. These values are then used by en-route filtering and selective forwarding of the messages carrying a matching content, once released by the publishers. This preliminary work led to the proposal of a scalable mediation service addressing the issues of information overload and a filter-based scheme for content dissemination in publish/subscribe systems. Some of those contributions were proposed in the design of a generic routing framework for Autonomic Networks.

Chapter B.1: Leveraging replication in content centric networking

> (Publication [6]: P. Spathis, S. Belouanas, K. L. Thai, "Leveraging Replication in Content-Centric Networks." *Global Information Infrastructure and Networking Symposium (GIIS*, 2015.)
> (Publication [7]: S. Belouanas, K. L. Thai, P. Spathis, M. Dias de Amorim, F. Rousseau, A. Duda, "Content Centricity in Constrained Cellular-Assisted D2D Communications." *Ad Hoc Networks*, 2014.)
> (Ph.D. thesis of Salah Belouanas, UPMC Sorbonne Universités, 2017. Co-supervised by Marcelo Dias de Amorim.)

In Chapter B.1, I present a scalable on-demand routing protocol that takes advantage of the paradigm of content-centric networking to meet the needs of content related services and data-intensive applications. This protocol exploits the high availability of content in large-scale data delivery networks. It complements the CCN routing framework with the ability to discover copies of replicated content in a scalable fashion. The novelty lies in the use of the caching capabilities of content routers to store the most efficient routes toward those copies.

Chapter B.2: Leveraging delay tolerance in cellular networks

(Publication [7]: S. Belouanas, K. L. Thai, P. Spathis, M. Dias de Amorim, "SCoD: Saving Cellular Resources by Delaying Transmissions of Popular Content." *12th International Symposium on Wireless Communication Systems (ISWCS)*, 2015.)
(Submission: S. Belouanas, K. L. Thai, P. Spathis, M. Dias de Amorim, "Leveraging Node Gatherings to Save Cellular Resources." *Annals of Telecommunications*, Springer, 2016).
(Ph.D. thesis of Salah Belouanas, UPMC Sorbonne Universités, 2017. Co-supervised by Marcelo Dias de Amorim.)

In Chapter B.2, I present SCoD (Scheduled Content Delivery), a new radio resource-saving strategy for disseminating content in cellular environments. This strategy takes advantage of the mobility of users who tend to group together and of their tolerance in delayed receipt of content. The aim of SCoD is to schedule the transmission of a given content in the cells hosting a sufficient number of users. By maximizing the benefit expected from a transmission in terms of receivers, SCoD succeeds in minimizing the number of transmitting antennas as well as the total number of transmissions required to satisfy all users.

Chapter B.3: Broadcast-based data dissemination in VANETs

(Publication [8]: R. A. Gorcitz, P. Spathis, M. Dias de Amorim, R. Wakikawa, and S. Fdida. "Dissecting dissemination in VANETs." *ACM 8th Asian Internet Engineeering Conference (AINTEC*, 2012.)
(Publication [9]: R. A. Gorcitz, P. Spathis, M. Dias de Amorim, R. Wakikawa, and S. Fdida. "Servus: Reliable low-cost and disconnection-aware broadcasting in VANETs." *IEEE 7th International Wireless Communications and Mobile Computing Conference (IWCMC*, 2011.)
(Ph.D. thesis of Raul A. Gorcitz, UPMC Sorbonne Universités, 2013 [10]. Co-supervised by Serge Fdida and Marcelo Dias de Amorim.)

Chapter B.3 focuses on the dissemination of information in vehicular networks. I present the two following contributions: A component-based analysis of the relevant dissemination protocols (Section B.3.1) and the design of Servus, a multihop broadcast-based protocol for reliable and robust data dissemination in vehicular networks (Section B.3.2).

Chapter B.1

Leveraging Content Replication

(Publication [6]: P. Spathis, S. Belouanas, K. L. Thai, “Leveraging Replication in Content-Centric Networks.” *Global Information Infrastructure and Networking Symposium (GIIS)*, 2015.)
(Publication [7]: S. Belouanas, K. L. Thai, P. Spathis, M. Dias de Amorim, F. Rousseau, A. Duda, “Content Centricity in Constrained Cellular-Assisted D2D Communications.” *Ad Hoc Networks*, 2014.)
(Ph.D. thesis of Salah Belouanas, UPMC Sorbonne Universités, 2017. Co-supervised by Marcelo Dias de Amorim.)

This chapter is concerned with the mismatch resulting of the host centric design of the Internet and the needs of content-intensive applications. I exploit the ability to access data objects in a location-independent fashion introduced by the content centric networking paradigm. I present the design of a routing scheme that takes advantage of the widespread availability of content in today’s large scale networks. More especially, this scheme takes benefits of content replication and caching as a common practice amongst service providers who can improve access efficiency by minimizing retrieval time while reducing bandwidth costs.

This work completes the existing communication framework of CCN [11] with a scalable inter-domain routing protocol able to discover routes that allow users to retrieve replicated content from efficient locations. We adopt an query-based approach for content discovery supported by the caching capabilities of content routers. To handle content replication, we borrow from the literature [12] the idea of using flat unstructured identifiers in replacement of CCN DNS-based names. To overcome the issue of scalability inherent to routing on flat names, we use the caching capabilities of content routers for storing content but also the routes discovered for popular content.

In summary, the contributions of this chapter are as follows:

- **Inter-domain content routing.** I describe a scalable discovery scheme of routes toward copies of popular content.
- **Route caching.** I show the benefit of caching the discovered routes in the content routers.
- **Flat namespace.** I introduce an unstructured location-independent naming space compliant with the replication or migration of content.

The remainder of this chapter is structured as follows. Section 1.1 provides the detailed description of our routing architecture including the namespace, the node model as well as the types of packets. The routing algorithm is presented in Section 1.2. Results of the simulation experiments are presented in Section 1.3.

1.1 | *One-to-nearest association in CCN*

A design objective common to most content-centric architectures is to improve content delivery performance and reduce traffic overhead. To achieve this objective, we claim that content-based routing should take benefit from the widespread availability of content in large scale networks as data objects may be replicated and cached at several places. By giving emphasis on the content rather than the location, content-centric networking renders binding of content to endpoints obsolete when providing access to the content. The paradigm underlying content-centric networking thus gives a natural way to make effective use of content replication since end-to-end connectivity to the source of a replica is not required prior to access the data. As such, content-centric networks naturally provide network-level support for discovery of replicated data items.

This section presents our on-demand inter-domain flat name-based routing protocol that integrates naming to routing for effective discovery of content replicas based on network metrics while providing resilience to network failures or content migration. Our approach overcomes the issue of scalability inherent to routing on flat names by making use of the caching capabilities of content routers for storing previously learnt routes.

The remainder of this section details how our architecture works. I first present the content identifier namespace, the node model as well as the data structures and the types of messages.

1.1.1 Content identifier

We introduce a flat namespace consisting of unstructured, location-independent, and human-unreadable identifiers called *Content Identifiers* (CIDs). The CIDs are said to be semantic-free since they provide the ability to refer to content and services independently of the hosting endpoint regardless of its administrative domain, network location, or the network topology. In contrast to the CCN DNS-like names, content can be replicated or migrate between hosts while avoiding broken links or using HTTP redirection which requires control of former domain, increases traffic, and latency.

The value of CIDs might be a 128-bit long number chosen randomly or could be derived from the cryptographic hash of the public key of the content owner and the content itself for authentication and integrity respectively. The granularity of naming is flexible since a name can refer to a host, a content within it, or at finer granularity any content item with no reference to the hosting endpoint. At the user-level, it is assumed that a mapping service provides applications with the CIDs corresponding to human readable descriptors such as search keywords which in this case, will require the mapping service to be carried out by search engines.

1.1.2 Message Types

CCN has two types of messages: Interest and Data messages. An Interest message is used to request data by name. A Data message is used to supply data. In CCN, the Interest messages contain the DNS-based name prefixes of the content items to be retrieved while the requested content is carried in a Data message identified by a name matching the prefix provided in the Interest. In our routing scheme, the Interest and Data messages now contain a content identifier (CID) field instead of the CCN DNS-based names.

We also extend both types of messages with a path-vector field borrowed from BGP update messages. This new field contains a path-vector similar to the BGP AS Path attribute which includes the list of Autonomous System Numbers (ASNs) that describes the sequence of routing domains through which the Interest has passed. The path-vector is used to prevent Interests from looping and is returned to the querier in the Data message carrying the requested content item. All subsequent Interests will be directed along this sequence of domains toward the routing domain where the content was found.

The path-vector is also used by the content routers to aggregate the Interest messages requesting the same content. A content router stores the shortest path vector followed by previous occurrences of an Interest and only forwards a subsequent one if received along a shorter route. An Interest message is also scoped by a time to live (TTL) which is initialized to the maximum number of routing domains the Interest message can visit.

1.1.3 Data Structures

We complete the operation of a CCN node by introducing an additional data structure to the CCN node model. In addition to the three existing structures, namely the Forwarding Information Base (FIB), the Content Store (CS), and the Pending Interest Table (PIT), we introduce a Route Store (RS). The FIB is used to forward Interests toward the sources having a copy matching the requested content. The CS is a cache memory where content objects can be stored to satisfy future Interests. The PIT keeps track of the interfaces on which Interests for a specific content are forwarded. We extend the PIT entries with information collected from the Interest header fields.

The RS is used to keep in cache the routes discovered in response to previous Interests. Each RS entry contains a path-vector representing the list of routing domains along the route. When a content router (CR) starts searching the best content replica with respect to its relative location, the CR sends an Interest either along the unicast path provided by BGP or will alternatively broadcast the Interest to all its peers in the neighboring routing domains depending on the popularity of the content. If a route for the requested content is present in the RS, the CR will use this route to direct the Interest towards the domain where a replica of the content has been previously found.

We extend the PIT entries in order to avoid propagating multiple instances of the same Interest. Each PIT entry is extended with the identifier of the CR who initiated the search, the sequence number of the Interest, and the shortest path vector followed by previous occurrences of this Interest sharing the same originating CR and sequence number. As in CCN, the PIT is also used to aggregate multiple Data messages so as only a single copy will be forwarded towards the content requester.

Takeaway B.1

To support the efficient discovery of replicated content, CCN DNS-like names are replaced by a location-independent flat identifiers. The format of CCN messages and the CCN node model are updated to accommodate the use of a BGP-like path vector for loop prevention and search aggregation. A path vector contains the best discovered route for a given content. The routers can either cache a copy of the content or the route itself.

1.2 | *Routing on content identifiers*

In this section, we describe how the design of our name-based inter-domain routing protocol supports the efficient discovery of some content items by using the plurality of alternative routes to access replicated content.

In contrast to CCN, Interests are propagated between neighboring domains in hope of finding the closest content replica. Our choice to avoid designing an advertisement-based protocol is motivated by the need of resiliency to content migration and network failures. Relying on periodic advertisements of routing information is expected to increase the network overhead when transposed to highly replicated content.

1.2.1 Domain

We refer to a *domain* as the administrative entity corresponding to a routing domain or an Autonomous System (AS). The boundary of a domain may also be defined by reflecting other organizational considerations related to the ownership of content available within the domain boundary. As such a domain may refer to a campus, a company, or a content provider.

A domain distinguishes content depending whether it is the originator of the content or a purveyor of a copy of the content. A domain containing a local copy of a content considers this content as an internal content. A *home domain* is a specific case since it contains the original copy of a content item provided by the content owner himself/herself. We assume that an intradomain routing protocol creates the routes needed by local content routers to forward the Interests along the shortest path towards an internal content.

A content item is considered as external to a domain if a copy cannot be found locally. In the absence of a local copy, a domain can still have a route previously learnt for this content. If not, a border content router (BCR) will initiate a search by sending the Interest to its neighboring domains in the same way BGP propagates its update messages. BCRs are configured to be BGP peers and as so, will establish a BGP session running on top of TCP prior to the exchange of the Interest messages throughout the search process. Inside each domain, a BCR is in charge of propagating the Interests requesting an external content.

1.2.2 Requesting an external content

As depicted in Figure B.1, when an Interest for a replicated content is received in a domain other than the home domain, the border content router (BCR) of this domain first checks if the path-vector included in the Interest contains its own domain ID in which case the Interest will be discarded. In the absence of loops, the BCR appends the local domain ID to the Interest path-vector and checks the PIT for an entry matching the incoming Interest, i.e. containing the same original requester and sequence number. A PIT-match indicates that a previous Interest has already been sent for the same content. The Interest is discarded if the path-vector included in the PIT entry is shorter than the one carried by the Interest.

Otherwise the PIT entry is updated with the Interest path-vector and arrival interface. If no PIT entry is found, a new PIT entry is created with the identifier of the BCR who initiated the search, the sequence number of the Interest, the path-vector it carries, and the interface on which the Interest arrived. The BCR then looks in its Content Store for a matching content. If found, a Data message carrying the content will be returned to the requester by following the PIT entries left behind the Interest.

In case of no Content Store match, the BCR checks in its Route Store (RS) if a route was learnt through a previous search. If such a route exists, the Interest is forwarded along this route towards the content replica. If no matching route can be found in the RS, the BCR decrements the TTL of the Interest message. If the Interest TTL did not reach zero, the BCR sends out the Interest to all its neighboring peers or along the unicast BGP route depending on the popularity of the content.

A timer is associated to each PIT entry to limit the rate of Interests and to handle unsuccessful searches. If no matching Data is received, the original requester is responsible for generating a new Interest different from previous occurrences thanks to an incremented sequence number.

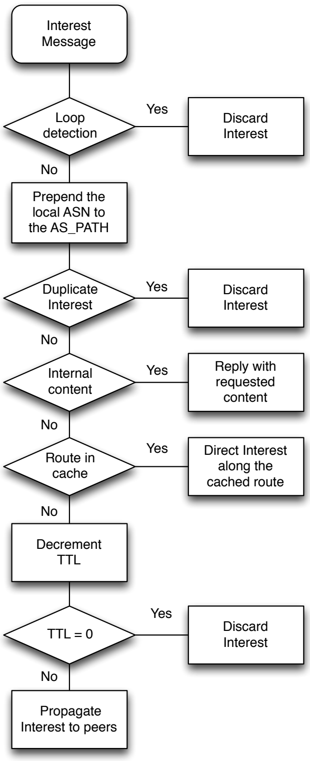
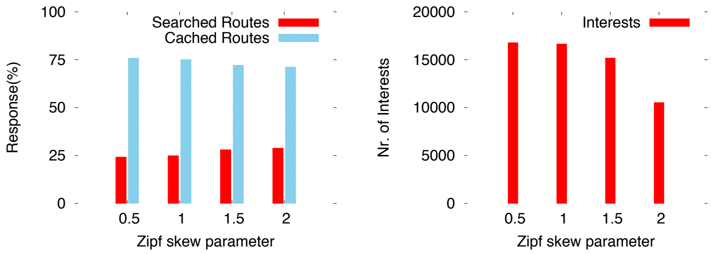


Figure B.1. Handling Interests.

1.2.3 Responding with an internal content

Upon the receipt of an Interest for an internal content, a border content router (BCR) starts performing loop-detection on the Interest path vector in a similar way to BGP. The BCR will append the ID of its domain to the path-vector if loop-free. The resulting sequence of domains corresponds to the complete route between the originating BCR that initiated the search and the local replica of the content. This route is sent back to the originating BCR along with the copy of the local replica in a Data message. The Data message also inherits the Interest sequence number and includes the identifier of the replying BCR.

Upon receiving the Data message, a content router (CR) extracts the path vector contained in its header and stores the corresponding route in its Route Store along with the identifier of the BCR who originated the Data message. This path vector will be used to direct subsequent Interests requesting the same content. The CR also stores the content object carried by the Data message in its Content Store before forwarding the Data message on every interfaces listed in the



(a) Discovery process benefit.

(b) Search messages overhead.

Figure B.2. Content popularity.

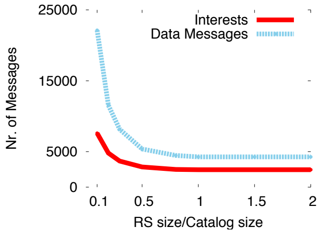


Figure B.3. Content catalogue.

PIT entry left by previous Interests. The PIT entry is erased once the Data message forwarded as a way to prevent subsequent copies of the content from being propagated later.

The route contained in the Data message will be advertised within the domain of the original requester so that internal content routers may also learn how to reach the external content in case of subsequent Interests. This route will also be used so as to handle Interests received from external peer domains.

1.2.4 Default routing for sparse content

Since some content can exhibit a low degree of replication, we introduce an alternative using the unicast route towards the home domain of the requested content. In order to enable the use of CIDs for unicasting Interests, we use the multiprotocol extensions of BGP (i.e. MP-BGP) [13] to carry routing information for the CCN layer using CIDs.

We propose to define a new address family called CCN AF used to identify CCN as the network layer protocol carried by BGP. To make BGP advertise a set of CIDs as reachable destinations together with the next hop identifier to be used for forwarding along the feasible routes to these destinations, we use the CCN AF in combination with the two BGP attributes Multiprotocol Reachable NLRI (MP_REACH_NLRI) and Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI) introduced to enable BGP multiprotocol extensions capability. Both attribute contains one or more triples that consists of the Address Family information, the Next Hop Information, and the Network Layer Reachability Information (NLRI).

In our protocol, the Address Family information identifies CCN as the network layer protocol that is being carried within the update. The Next-Hop Information is the identifier of the next border CCN router on the path to the destination. In our protocol, the NLRI is encoded as one or more 2-tuples with the following format: Length (1 octet): Total length of the CID advertised; CID: The content identifier.

Takeaway B.2

The copies of a content can either be found within a stub domain or in the cache of the routers. To discover such a copy, CCN Interest messages are exchanged between neighboring domains or follow the routes already known to the routers as a result of a previous search. For unpopular content, the Interest messages follow the unicast routes of the content original copy.

1.3 | *Performance evaluation*

1.3.1 Experimental setup

In order to evaluate our proposal, we developed *ccnSim*++, an extension to the *ccnSim* simulator [14]. *ccnSim* is a chunk-level simulator built in C++ and developed on top of the *Omnet*++. Its ability to scale well in dense topologies allowed us to conduct our simulations on a large dataset made publicly available by the Cooperative Association for Internet Data Analysis (CAIDA). This dataset consists of a snapshot of more than 36,000 ASes and around six million announced BGP routes. In our simulations, we assume that all links have unlimited bandwidth and are not affected by congestion. All our tests were performed on a machine equipped with 144GB of RAM and a 24-core Intel Xeon processor.

1.3.2 The influence of popularity

In Figure B.2a, we show the percentage of routes that are actually discovered when searching for a content or reused from a previous successful search as a function of the popularity of the content which follows a Zipf distribution. We can see that over than 75% of the routes used to retrieve a content are provided from the Route Store of the content routers. This shows the benefit of caching the routes discovered toward a replica as a result of requesting a content. The overhead of the discovery process is outweighed by the benefit of caching the routes to the replicas we discover. The benefit of caching routes toward previously discovered replicas can also be found in the number of Interest messages.

Figure B.2b shows the total number of Interest messages forwarded in the network as a function of the popularity of the content. Though the number of Interest messages issued for a content increases with the popularity of the content, we can see an opposite behavior when we increase the Zipf skew parameter. This indicates that Interest messages travel less distance in the network to discover a content as its popularity increases. Interest messages can draw on previous searches as content routers intercept them and reply with cached routes learnt as a result of previous searches. It is also expected that the Data messages sent on a route fetched from a Route Store will travel along a shorter path, reducing this way the downstream bandwidth utilization and improving the latency for users retrieving the content.

1.3.3 Route Store and catalog size

With a large catalog size, the route store is unable to hold all the best routes to the popular content. In Figure B.3, we evaluate the impact of the size of the content catalogue as we expect the limited storage of Route Stores to impact on the performance of our route discovery scheme. We vary the size of the Route Stores for different sizes of the content catalogue. We measure the resulting number of Interest and Data messages as a function of the ratio of the RS size over the catalogue. We can observe that for small increases in the size of the Route Stores, the number of messages generated by our architecture decreases significantly. For size ratios greater than 0.8%, this number of Interest and Data messages remains stable. This is due to the fact that the size of the route stores becomes large enough in comparison to the catalogue size to hold the routes discovered for all popular contents.

Takeaway B.3

The caching capabilities of the content routers can mitigate the overhead resulting from the discovery of popular content in CCN. Storing the routes previously discovered toward content replicas help reduce the number of Interests but also the distance they travel. The improvements resulting of caching routes in addition to the content objects are significant even in the face of large content catalogues without adding more requirements on the memory size of the content routers.

1.4 | Conclusion

This chapter addressed the mismatch resulting from the host centric design of the Internet with regard to the needs of content related services. We proposed a scalable on-demand routing protocol that takes advantage of the paradigm of content centric networking.

As part of our design process, we take advantage of the high availability of content in large-scale networks. The novelty of our approach lies in the use of the caching capabilities of content routers to store the routes discovered toward content replicas. Our protocol supplements the CCN routing framework with the ability to discover efficient routes that allow users to retrieve replicas in a scalable fashion. In the next chapter, we take account of the user need in accessing data while on the move.

Chapter B.2

Leveraging Mobile User Gatherings

(Publication [7]: P. Spathis, S. Belouanas, K. L. Thai, “SCoD: Saving Cellular Resources by Delaying Transmissions of Popular Content.” *12th International Symposium on Wireless Communication Systems (ISWCS)*, 2015.)
(Submission: S. Belouanas, K. L. Thai, P. Spathis, M. Dias de Amorim, “Leveraging Node Gatherings to Save Cellular Resources.” *Annals of Telecommunications*, Springer, 2016).
(Ph.D. thesis of Salah Belouanas, UPMC Sorbonne Universités, 2017. Co-supervised by Marcelo Dias de Amorim.)

In this chapter, I address the capacity limitation that cellular operators are facing due to the increasing demand of users for data while on the move. The work I present is concerned with cellular wireless networks where mobile users are covered by a collection of antennas. I propose to reduce the consumption of the radio resources by controlling the number of transmissions, whenever possible as a way to remove redundant traffic. This work is motivated by the following simple hunch: Determining *when* to transmit could lead to significant savings in situations where data is popular and tolerates some delay.

The main contribution of this work came with **SCoD (Scheduled Content Delivery)**, a radio resource-saving strategy for disseminating delay-tolerant content in cellular environments. SCoD draws on the mobility of users to schedule a multicast transmission whenever a cell hosts a sufficient number of users. Given a set of wireless antennas covering cells spanning a population of users interested in a content item, *the objective of SCoD is to compute the time(s) each antenna should transmit the content so as to satisfy all interested users*. SCoD determines within each cell, which number of users should trigger a transmission so as to reduce the overall number of transmitting antennas and, as a consequence, the total number of transmissions required to satisfy all users requesting the same content. This is especially challenging if no prior knowledge of user mobility is available.

The contributions of this chapter are as follows:

- We propose a decision function that determines whether the number of users within the same cell is sufficient to trigger a transmission. Such a function proceeds by periods that users tolerate before a missing content is considered as a failed transmission. The typical behavior of the decision function is as follows. At the beginning of a period, SCoD seeks for large gatherings of users within each cell. As time goes, SCoD gets more pragmatic and relaxes the requirement on the size of user gatherings. The bullishness of SCoD in the number of users required to trigger a transmission fades toward the end of a period.
- We evaluate the performance of SCoD using various heuristics and compare it with two alternative strategies. The first corresponds to “no sooner requested than transmitted” strategy. We will refer to this strategy as the Hot Potato strategy in the rest of this chapter. The second strategy operates under the assumption of an Oracle, which knows all the future movements of users. Though the Oracle strategy is not realistic in practical settings, it provides us an upper bound regarding the benefit of delayed transmissions.

2.1 | *System model and problem statement*

We consider a system consisting of a group of users scattered across a geographical area covered by a collection of access points. The bottleneck of this system results from the limited capacity of the wireless access channel. We suppose content to be disseminated as a result of a publish/subscribe system where users subscribe to a content. Instead of pushing the content once available, we are interested in measuring how long a content can be delayed so to maximize the number of subscribers within the same cell.

Model definition. Our system is defined by the tuple $[c, \mathbf{N}^c, \mathbf{A}, t_0^c, \tau]$, where c is the content to be disseminated, $\mathbf{N}^c$ the set of users interested in receiving c, $\mathbf{A}$ the set of access points (i.e. cells), t_0^c the time when c becomes available, and τ the maximum delay that all users tolerate before receiving c. We will refer to time $t_0^c + \tau$ as the content deadline of c. We assume that content c remains valid during the interval $[t_0^c, t_0^c + \tau]$.

For the sake of clarity, we consider that periods of dissemination do not overlap, meaning that there is one and only one content present in the system at a time. This assumption does not impact our conclusion, as we evaluate the system in a per-content basis.[1] Also, we consider that when a content expires, another one is generated.

[1] In fact, we derive a lower bound on the gains that we can obtain with SCoD, as multiple content would benefit from the same user gatherings, leading to additive gains. Dealing with multiple content in the system at the same time will be subject of future work.

Unless specified, we do not deal with any specific radio technology and will simply refer to $a_i \in \mathbf{A}$ as a wireless access point (or a base station) that covers a geographical area called *cell*. Cells do not overlap and cover the whole target zone. It is equivalent to say that a user is covered, at all times, by one and only one access point at a time. We assume, however, that the cellular infrastructure transmits using multicast. In other words, a single transmission is enough to push the content to all interested nodes in the respective cell. Thus, in the rest of this chapter, whenever we use the term transmission we mean multicast transmission.

We finally assume that the population $\mathbf{N}^c$ remains unchanged during τ. Nodes in $\mathbf{N}^c$ belong to either one of the two following subsets: (*i*) $\mathbf{N}^{c\oplus}_{a_i}(t)$, the set of users associated with access point a_i at time t that have not yet received the content and (*ii*) $\mathbf{N}^{c\ominus}_{a_i}(t)$, the set of users associated with access point a_i at time t that have already received the content. Hence, $\mathbf{N}^c = \bigcup_{i=1}^{|\mathbf{A}|}[\mathbf{N}^{c\oplus}_{a_i}(t) \bigcup \mathbf{N}^{c\ominus}_{a_i}(t)], \forall t \in [t_0^c, t_0^c + \tau]$.

We define a *configuration* at time t as a snapshot of the graph consisting of the clusters of users associated with each access point. A configuration changes whenever at least one user moves to another cell. A single transmission by an access point is received by all users in the corresponding cell. Each access point knows at all times the list of mobile users it covers.

Problem statement. Our goal is to reduce the number of transmissions by taking into account the mobility of the users, on the one hand, and the delay tolerance of content, on the other hand. In other words, given the tuple $[c, \mathbf{N}^c, \mathbf{A}, t_0^c, \tau]$, we seek to find the minimum set of pairs (t_i, a_i) that covers all nodes interested in receiving c, where $t_i \in [t_0^c, t_0^c + \tau]$ is the time access point a_i transmits a copy of c. Given the mobility of users which may not be known in advance, transmitting a content as soon as available may lead to inefficient transmissions while transmitting as late as tolerated by the users may result in missing opportunities of reducing the total number of transmissions.

Takeaway B.4

Delaying even slightly the transmission of a popular content can reduce the consumption of the radio resources in the context of a cellular network. The delay tolerance of content combined with the natural tendency of users to get together in the same locations can help reduce the total number of transmissions required to satisfy all users requesting a given content.

2.2 | *Scheduled content delivery*

We propose SCoD (Scheduled content delivery), a strategy that decides when to trigger a transmission based on both the number of nodes that have yet to receive a content c and on the remaining time before the content deadline. In a nutshell, SCoD first checks whether the deadline has been reached or not. If so, the decision is to multicast c in every cell containing at least one node still waiting for c. If not, SCoD checks if the number of nodes within each cell is sufficient to trigger or not a transmission at time t.

Triggering transmissions. Formally speaking, access point a_i transmits content c at time t if one of the two conditions below holds:

$$\begin{cases} |\mathbf{N}^{c\oplus}_{a_i}(t)| \geq \omega\beta(t) \sum_{j=1}^{|A|} |\mathbf{N}^{c\oplus}_{a_j}(t)|, & \forall t \in [t_0^c, t_0^c + \tau[, \\ |\mathbf{N}^{c\oplus}_{a_i}(t)| > 0, & t = t_0^c + \tau, \end{cases} \tag{2.2.1}$$

where $|\mathbf{N}^{c\oplus}_{a_i}(t)|$, as previously mentioned, is the number of nodes gathered in the coverage of access point a_i at time t and interested in content c, but yet to receive c. Function $\beta(t)$, called *lateness function*, defines the tolerance of SCoD regarding the size of a gathering. SCoD is less tolerant in the beginning of the period, i.e. it requires bigger groups of nodes in a cell. As time goes by, SCoD becomes more realistic as $\beta(t)$ tolerates smaller groups. Once the content deadline $t_0^c + \tau$ reached, c is sent in all cells containing at least one node who has yet to receive c.

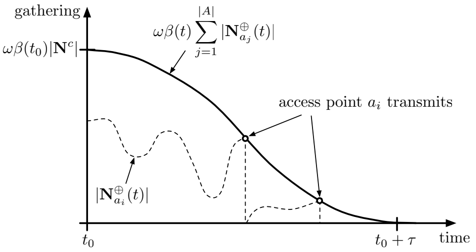


Figure B.1. Conditions for SCoD to trigger transmissions.

The lateness function $\beta(t)$ is monotonically decreasing and normalized over the interval $[0, 1]$, i.e. $0 \leq \beta(t) \leq 1, \forall t \in [t_0, t_0 + \tau]$. As shown in Equation 2.2.1, $\beta(t)$ is associated with a parameter ω that determines the *aggressiveness* of the decision process in the beginning of the interval. In fact, $\omega \in [0, 1]$ is a weighting coefficient that reflects the dissemination's requirements level. If $\omega = 1$, it means that content c is transmitted only if, at the beginning of the dissemination period, 100% of the nodes gather under a single access point. Similarly, $\omega = 0.5$ means that at the beginning of the dissemination, we need to have at least 50% of the population $\mathbf{N}^c$ gathered under the same access point to trigger a transmission. In Figure B.1, we illustrate the operation of SCoD. The bold line is the envelop provided by SCoD while the dashed line gives the number of mobile nodes associated with access point a_i and that have yet to receive the content. A transmission is performed when the two curves meet.

Determining ω and $\beta(t)$. The efficiency of SCoD depends on the proper choice of both parameter ω and the lateness function $\beta(t)$. Indeed, if ω is too close to one, we may never trigger any transmission if nodes do not tend to gather under the same access point. On the other hand, if ω is too loose, SCoD will trigger an instant transmission and no improvements will be observed. Furthermore, the decaying behavior of $\beta(t)$ will dictate how much SCoD should relax its requirement regarding the gathering level in each cell during the dissemination interval.

In practice, as node mobility cannot be known in advance, we need to rely on past observations to set the parameters of SCoD. To this end, we collect during each τ the maximum number of users gathered at the same time under the same access point (we call this the *gathering ratio*). We use the most recent observation of this number to set the value of ω. Given that each interval concerns a specific content, the group of nodes differs from one interval to another. We divide the total number of users by the number of interested users in order to have a normalized value.

For the sake of illustration, we compute this ratio using the Bologna dataset [15] with a varying number of access points (100, 200, 300, 400, and 500). We assume $\tau = 300$s. The results are shown in Figure B.2. This figure represents the values of the gathering ratio for consecutive intervals of length τ. Although the ratio variation is somehow erratic, the variations of the ratio from one interval to the next are not significant in most of the cases. This is even more true for larger values of τ. These results indicate that we do not need to look back further in the past – observing the previous interval seems to be enough to have an idea about how the population is likely to gather in the next interval.

For the lateness function $\beta(t)$, we adopt a similar approach. We test several functions on a recent observation over a window of size τ and reuse the same function (together with the corresponding ω) during the current interval. In practice, we consider three different monotonically decreasing functions in our tests. They are shown in Table B.1.

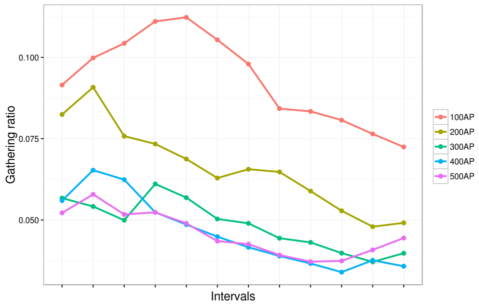


Figure B.2. Variation of the gathering ratio for consecutive intervals with duration τ = 300s.

Takeaway B.5

SCoD takes the decision to transmit a content depending on the number of interested users within the same cell. A lateness function defines for each antenna the transmission times without prior knowledge of user mobility. This function determines how many users should be gathered within a same cell before triggering a transmission.

Function	Expression
Linear	$\beta(t) = \frac{-(t-t_0)}{\tau} + 1$
Square-root	$\beta(t) = \sqrt{\frac{-(t-t_0)+\tau}{(t-t_0)+\tau}}$
Power	$\beta(t) = (\frac{-(t-t_0)}{\tau} + 1)^3$

Table B.1. Lateness functions used for $\beta(t)$.

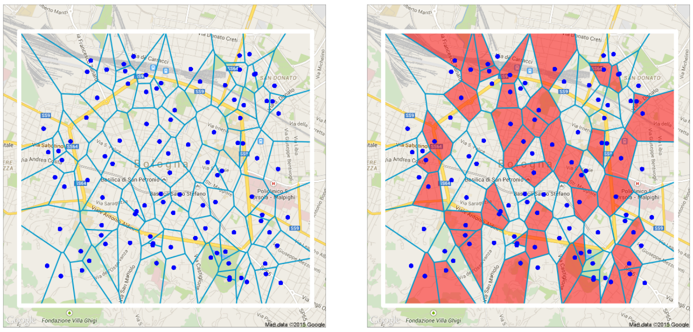


Figure B.3. (a) The Voronoi diagram for a set of 100 access points deployed in the city of Bologna. (b) The red cells are the one involved in the transmission process of a content.

2.3 | *Dataset and transmission strategies*

To evaluate our proposal, we use the Bologna vehicular dataset [15]. This dataset contains a significant number of mobile nodes (more than 10,000 vehicles) dispatched all over the city, which corresponds to about 10 km^2. We track the location of the vehicles for a period of one hour.

The Bologna dataset describes the real movements of vehicles but provides no information on the connectivity these vehicles may have with the Internet. To this end, we emulate a cellular network using a Voronoi diagram that covers the entire city of Bologna and where each Voronoi cell contains only one access point [16]. This gives us some flexibility to evaluate our proposal under multiple density conditions and handoff frequencies. As an illustration, we depict in Figure B.3 the Voronoi tessellation of 100 access points within the central area of Bologna.

To give more confidence to our results, we simulate multiple deployment plans following a uniform distribution. We generate 10 different topologies for each number of access points $|\mathbf{A}|\in \{100, 200, 300, 400, 500\}$, which makes 50 scenarios in total. In regard to the content deadline, we consider several scenarios with the following values of $\tau \in \{60s, 120s, 180s, 300s\}$. The results that we will present in the next section are the average of 10 simulation runs conducted for each topology size.

2.3.1 Transmission strategies

We compare SCoD with the two following strategies: Hot Potato and Oracle.

Hot Potato. This strategy transmits a content as soon as available, wherever the subscribers are. Using the notation we introduced in Section 2.1, access point $a_i \in \mathbf{A}$ transmits a copy of content c at $t = t_0^c$ only if $\mathbf{N}_{a_i}^{c\oplus}(t_0) \neq \emptyset$ (i.e. only if at least one subscriber is in cell a_i). As a result, the number of transmissions is equal to the number of cells where subscribers are located at t_0^c. The Hot Potato strategy is agnostic to the mobility of users and to their tolerance in receiving a delayed content.

Oracle. In contrast to the Hot Potato strategy, the Oracle strategy relies on the complete *a priori* knowledge of user mobility for the entire period τ. This strategy draws on this knowledge to determine precisely which access points should transmit and at which points of time so to maximize the benefits of each transmission. The Oracle strategy takes full advantage of the tolerance of subscribers in receiving a content with a slight delay to make the best out of each transmission in terms of subscribers satisfied.

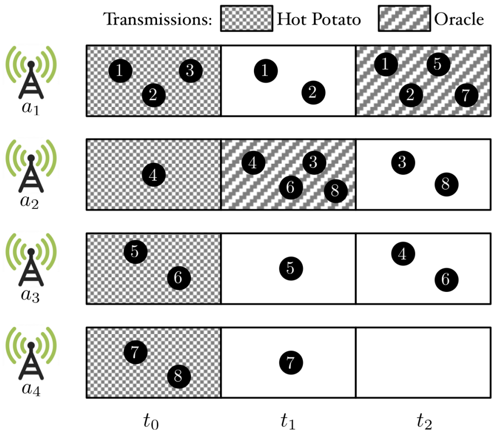


Figure B.4. The Hot Potato strategy transmits as many copies of content than the number of cells containing subscribers at $t = t_0$. Since all four cells contain at least one subscriber at t_0, Hot Potato requires four transmissions. The Oracle strategy, on the other hand, covers all eight subscribers with only two transmissions.

Finding the best timing for transmitting amounts to solve the well-known minimum set cover problem [17] which is also known to be NP-hard. We use a simple greedy algorithm that provides an acceptable approximation of the best solution in a reasonable time [18, 19, 20]. We list the sequence of configurations describing the association between subscribers and each access point over period τ. We then select the minimal sublist of configurations covering all subscribers. The content is then transmitted to the precomputed sets of nodes that will lead to an approximation of the optimal solution.

Figure B.4 shows a simple example where we consider a population of $|\mathbf{N}^c|= 8$ users and $|\mathbf{A}|= 4$ access points. All eight users are interested in receiving a content c made available at $t = t_0$. The deadline for transmitting c is set at $t_0 + \tau$. We discretize time by dividing interval $[t_0, t_0 + \tau]$ into three time slots defining four discrete points of time t_i with $i \in [0, 3]$. A point of time t_i reflects a new *configuration* of the network. We will consider, without loss of generality, that all subintervals are equal to one second.

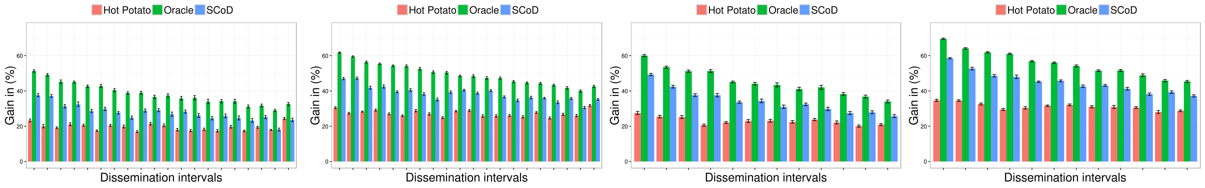
Since Hot Potato transmits the content as soon as available, four transmissions are required to cover all 8 subscribers, each of the four cells a_1, a_2, a_3, and a_4 hosting at least one subscriber at point of time t_0. On the other hand, Oracle leverages delayed content distribution to determine the minimum set of transmissions. As a consequence, Oracle requires only two transmissions (cells a_1 and a_2 at point of times t_2 and t_1, respectively).

Takeaway B.6

The Bologna dataset gives the location of more than 10,000 vehicles circulating through Bologna. The placement of a varying number of antennas is obtained by partitioning the city of Bologna following a Voronoi tessellation. We compare the performance of SCoD against two baseline strategies with different requirements regarding the knowledge of the node mobility.

2.4 | Evaluation results

To assess the benefits of SCoD, we compare SCoD with the two strategies presented in Section 2.3.1: Oracle (O) and Hot Potato (HP). To this end, we define a baseline strategy that we call “Full Coverage” (FC), where the infrastructure systematically distributes the content in every cell that have received the visit of at least one interested user during the



(a) |**A**|= 300 AP, τ = 180s. **(b)** |**A**|= 500 AP, τ = 180s. **(c)** |**A**|= 300 AP, τ = 300s. **(d)** |**A**|= 500 AP, τ = 300s.

Figure B.5. Average transmission gain for SCoD, our proposed strategy, compared with two baseline strategies (Hot Potato, Oracle). We use different values for the content deadline and access point configuration. Confidence intervals of 95% are shown as well.

interval $[t_0^c, t_0^c + \tau]$. This strategy is clearly too naive to be implemented and serves only as a lower bound for our evaluation.

To assess the benefits of SCoD, we compare it with the other strategies as described in the previous section. We will see that our proposal performs well in saving wireless resources by decreasing the number of transmissions. Additionally, we point out the efficiency of our system in terms of adaptability upon the arrival of a new content.

Average gain. We consider the *gain* $\mathcal{G}$ to compare the solutions. This parameter measures the savings that a given solution provides when compared with the Full Coverage strategy. It is formally defined as:

$$\mathcal{G}_{\text{strategy}} = \frac{N_{\text{FC}} - N_{\text{strategy}}}{N_{\text{FC}}}, \tag{2.4.2}$$

where N_{FC} is the number of transmissions when the Full Coverage strategy is used and N_{strategy} is the number of transmissions for the "strategy" under consideration.

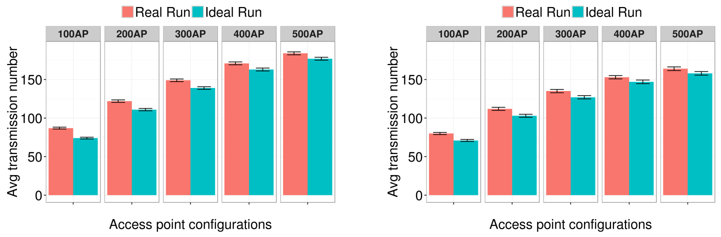
We plot in Figure B.5 the average transmission gain as defined in Equation 2.4.2. Each point in the x-axis represents an interval of duration τ – this means that each bar in the plot corresponds to the dissemination of one content. The y-axis gives the gain $\mathcal{G}_{\text{strategy}}$ on a per strategy basis.

We show the results for four combinations of $|\mathbf{A}|$ and τ. We observe that SCoD outperforms the Hot Potato strategy in almost all cases (except in two intervals when $|\mathbf{A}|$=300 and τ=180s). Also, the later the content deadline, the bigger the gains. This is inherently related to the rationale behind our strategy, where we take benefit as much as possible of the mobility of the nodes. In fact, when the deadline increases, it gives more time to our strategy to adapt and wait for a better gathering of nodes.

We observe also that the topology size has an impact on saving up resources. As the number of deployed access points increases (i.e. smaller cells), user movements result in more frequent handoffs, which leads to fast-varying cell densities. SCoD exploits well such a phenomenon and achieves gains that get closer and closer to the best possible result (i.e. the Oracle).

Real run vs. ideal run. We are interested in evaluating the adaptiveness of our system. To do so, we compare the number of transmissions generated by our strategy in real time (*Real Run*) against the best case, which corresponds to the case where SCoD runs with the best possible weighting coefficients (*Ideal Run*). Of course, the ideal run is unfeasible in reality as it requires future movements to be known in advance. We also compare the average number of transmissions for our strategy against its ideal run case, i.e. if the most appropriate values of $\beta(t)$ and ω were known in advance. It shows how far the strategy is, compared with its own best possible performance.

We observe in Figure B.6 that the configuration size and the number of transmissions are strongly correlated – as expected: as we increase the number of access points, the number of transmissions increases as well. We also note that the difference between the Real Run and the Ideal Run is finally not that significant. This somehow confirms that using measurements from the latest interval is enough to parametrize SCoD and obtain good results.



(a) τ = 180s. **(b)** τ = 300s.

Figure B.6. Average number of transmissions of our proposal in two cases: the Real Run that happens in reality and the Ideal Run that uses the best possible weighting coefficients.

Takeaway B.7

The performance of SCoD improves with smaller sized cells and for content with longer deadlines. The evaluation results also show that SCoD performs comparably whether the knowledge regarding the node mobility is known in advance or predicated based on recent observations.

2.5 | Conclusion

We proposed SCoD, a new content dissemination strategy specifically designed for cellular networks that suffer from heavy capacity constraints at the wireless access channel. It leverages node mobility and delay-tolerance to determine the best possible instants at which an access point must transmit a copy of a given content to a multitude of receivers. In a nutshell, its goal is to let users gather within fewer cells to reduce the number of transmissions. Experiments using a realistic vehicular dataset in the city of Bologna reveal that SCoD manages to reduce the number of transmissions when compared with strategies that do not play with the mobility of the nodes, while achieving 100% content delivery.

Chapter B.3

Data Dissemination in VANETs

This chapter presents two contributions to the domain of Vehicular ad hoc networks (VANET). VANETs is an active area for research and development likely to become an integral part of the communication infrastructure of tomorrow. From road safety to recreation applications, a wide range of services are emerging from the rapid technological advances in vehicles sensing, computing, and wireless capabilities. The key issue with respect to those new purposes is the ability of vehicles to disseminate information, such as warning messages to nearby vehicles. Due to the specific technical challenges inherent to vehicular environments, a wide variety of protocols have been proposed to cope with specific application requirements and environment constraints at the cost of increased complexity. My two contributions to this domain are the following:

- **Component-based analysis** (Section 3.1) I present a state of the art survey of broadcast-based dissemination protocols in VANETs. This survey provides a breakdown of the communication functions of the significant protocols in a collection of conceptual components. I present the results of a comparative evaluation of the various technical realizations proposed for each component.

- **Servus** (Section 3.2) The second contribution consists in the design of Servus, a broadcast-based dissemination protocol that efficiently tackles the challenges inherent to the VANET environment including intermittent connectivity. Servus successfully disseminates data messages to all vehicles located within well-defined areas without relying on roadmap data. Servus mitigates broadcast storms by avoiding message repetitions or rebroadcasting the same message multiple times.

3.1 | *A component-based analysis of VANETs dissemination protocols*

> (Publication [8]: R. A. Gorcitz, P. Spathis, M. Dias de Amorim, R. Wakikawa, and S. Fdida. "Dissecting dissemination in VANETs." *ACM 8th Asian Internet Engineeering Conference (AINTEC*, 2012.)
> (Ph.D. thesis of Raul A. Gorcitz, UPMC Sorbonne Universités, 2013 [10]. Co-supervised by Serge Fdida and Marcelo Dias de Amorim.)

To support the demand for data dissemination in VANETs, new functionalities are continuously introduced through the proposal of several protocols at the cost of increased complexity. Each proposal is designed to cope with specific application requirements and environment constraints in a more or less optimal fashion. Although the actual implementations of these dissemination protocols differ significantly in many aspects, a few common conceptual components exist over the whole range of protocols.

The component-based analysis I present in this section consists in a breakdown of the communication functions in a generic set of elementary building blocks, namely (*i*) neighbor discovery, (*ii*) broadcast containment, (*iii*) loss recovery, and (*iv*) disconnection tolerance. For each component, we examine their respective technological realization. By analyzing the implementation of the corresponding communication mechanisms, we propose a classification of the corresponding approaches into well-defined categories. In this way, we can better understand the behavior of each component and the interactions that result from their combination. We also identify which realization can best fit to different application profiles or communication scenarios giving the environment parameters, while having regard to the performance cost.

3.1.1 Component breakdown

The analysis of the significant broadcast-based protocols for disseminating information between vehicles lead to the identification of four atomic components:

- **Neighbor discovery.** This component helps retrieve information about the vehicle surrounding environment. It relies on sending periodic beacon messages.
- **Broadcast containment.** This component improves the overall network performance by avoiding or discarding redundant messages. It avoids the so-called broadcast storm problem that arises with blind flooding.
- **Loss recovery.** This component makes the dissemination of data reliable by repairing losses. It fits the requirements of critical safety messages regarding emergency brake or weather condition.
- **Disconnection tolerance.** This component overcomes the problem arising when the dissemination of a message is stopped because of connectivity gaps. A message is stored until forwarded when new neighbors become reachable.

3.1.2 Component evaluation

In this section, I give a classification of the different technological realizations proposed in the literature for each of the four aforementioned components. I choose one representative realization of each category and I compare the performance of those realizations. Each component is evaluated with regard to the node mobility, node density, and broadcast traffic overhead.

I first describe the simulation environment and then analyze the performance results obtained in NS-2 for the different scenarios. Each component is evaluated according to specific simulation scenarios capturing various application profiles and environment constraints. We focus on simulating each component at the network layer since no assumption is required concerning the MAC layer. All evaluated protocols are designed to operate independently of the underlying technology. In our evaluation, we use 802.11 as the default MAC technology available in NS-2 network simulator.

Neighbor discovery. We identify two classes of technical realization for this component: the *periodic beaconing* approach [21, 22, 23, 9, 24, 25] and the *hybrid* approach which consists in piggybacking a beacon when other types of message are disseminated [26, 27, 28, 29, 30, 31].

Evaluation. We simulated the two approaches using a real map of the third district of Paris converted into a NS2 ready map. The mobility pattern of the vehicles is determined by trace files obtained with the TransLite application[1] built on SUMO (Simulator of Urban Mobility)[2]. We simulate the movements of 50 vehicles, each with a transmission range of 250 meters.

For periodic beaconing, the value of the beacon timer is chosen at random between 1 and 15 seconds. As expected, the hybrid approach greatly reduces the number of beacons as their number decreases as the frequency of the general purpose messages increases. We further investigate the impact of the beacon timer. We vary the duration of the beacon timer from 0.5 seconds to 1.4 seconds. We observed that with a value lower than 1 second, both approaches perform similarly. When the value of the hold timer is greater to 1 second, the hybrid approach reduces the number of beacon messages by a factor of 2.

Broadcast containment. We identify two classes of technical realizations for the broadcast containment component: the *selective* approach and the *all-out* approach. They have been designed to also ensure reliable information dissemination.

Selective approach. This approach consists in limiting the number of nodes in charge of rebroadcasting a message [32, 33]. A rebroadcasting node is chosen for being the furthest in the dissemination direction of the message. The rebroadcaster is selected among the neighboring nodes located in the transmission range of the previous one who computes the distance to all its neighbors. The selection process is confirmed by a DATA/ACK exchange between the selected node and the previous rebroadcaster.

All-out approach. This approach takes advantage of the positive acknowledgments (ACK) sent for reliability purposes [9]. As in the selective approach, the purpose is to select the furthest node among all of those who received a message and to let this only node further propagate the dissemination of the message. To this end, all nodes triggers a transmission timer. If no ACK is received before the timer expires, a node rebroadcasts the message. If an ACK is received from a neighbor located farther in the dissemination direction, the node stops its timer and discards the message. Positive acknowledgments avoid the traffic overhead resulting from simultaneous broadcasting.

We conduct the performance evaluation of those two classes of realization in the next section.

Loss recovery. The technical realizations of this component can be classified according to the same categories identified for the broadcast containment component.

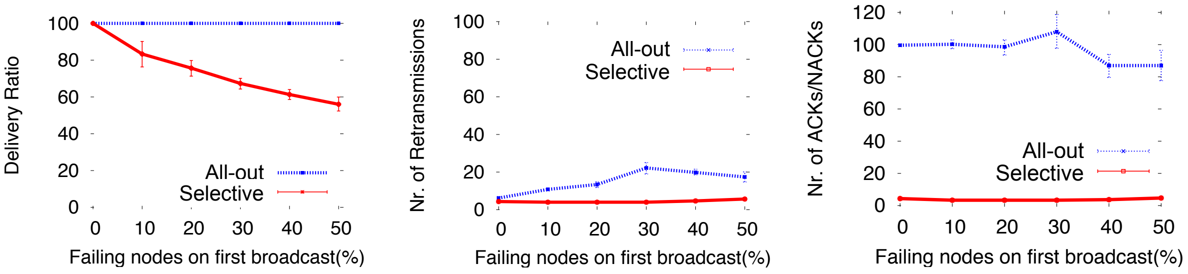
Selective approach. In this approach [32, 33], once the furthest node identified within the transmission range of a broadcaster, a DATA-ACK exchange is used between the two nodes to ensure reliable propagation of the dissemination. Reliability is thus guaranteed for the only sequence of nodes strategically located with respect to the dissemination latency and the broadcast containment.

All-out approach. In this approach [9], positive acknowledgement (ACK) are used to confirm correct reception of messages. After sending a message, a node starts a random retransmission timer. If no ACK is received before the timer expires, the node retransmits the message. On the receiving side, all nodes return an acknowledgment upon successful receipt of a message so to prevent the sender from retransmitting. If a node receives an unsolicited ACK for a message never received before, the node sends a NACK asking the nodes located ahead in the direction of travel to retransmit the missing message.

Evaluation. To evaluate the performance of those two approaches, we use a simulation scenario consisting of a 2,000-meter-long stretch of highway with traffic flowing in both directions. The total number of vehicles is set to 100 and

[1] http://lca.epfl.ch/projects/trans

[2] http://sumo.sourceforge.net/



(a) Delivery ratio.

(b) Retransmissions.

(c) Feedback messages.

Figure B.1. Disconnection tolerance and Reliability components.

each vehicle has a transmission range of 250 meters. We consider that all vehicles belong to a large cluster which has no transmission gaps. In our simulations, we evaluate the tradeoff between the reliability performance and the overhead incurred on the network. The reliability performance is analyzed in terms of delivery ratio while the overhead on the network is measured in terms of feedback messages (i.e. ACKs).

Figure B.1a shows the delivery ratio for the above two approaches as a function of the ratio of nodes that did not receive a message the first time it was broadcasted. We observe that the all-out approach achieves a higher delivery ratio even in the face of increasing delivery failures which indicates that all intermediate nodes belonging to the same cluster receive the report. With the selective reliability approach, the delivery ratio decreases as more intermediate nodes are left aside from the reliable dissemination process. Indeed the reliable delivery of the message is guaranteed for the only nodes selected as rebroadcasters.

The delivery ratio achieved by the all-out approach comes at the expense of an increased overhead. This is illustrated in Figure B.1c where we plot the number of feedback messages generated by both approaches. We observe that the amount of extra traffic following a loss remains constant with the selective approach. The all-out approach generates more control traffic since all vehicles, including the intermediate ones, are participating to the DATA-ACK exchanges. The same behavior is observed for the number of retransmissions as shown in Figure B.1b. When the ratio of nodes that did not receive a message at the first attempt exceeds a value of 30%, the amount of control traffic decreases before reaching a constant level as the all-out approach gives nodes the ability to exploit the unsolicited ACKs they receive from their neighbors. This ability allows the nodes to detect missing messages and request their retransmission. The same singularity can be observed in the results regarding the number of retransmissions.

Compared to the all-out approach, a lower overhead is incurred on the network with the selective approach. We can observe a tradeoff between the delivery ratio and the feedback overhead. As shown in Figures B.1c and B.1b, the number of feedback messages remains constant as reliability is only enabled for the nodes that will help cover larger areas and decrease the dissemination latency.

Disconnection tolerance. In this section, I present the three approaches of the intermittent connectivity component which will be referred to as *proactive*, *periodic*, and *reactive*.

Proactive approach. This approach consists in using the vehicles moving in the opposite lane of traffic to disseminate the warning messages upstream without adding broadcasting overhead. Here we are interested in the use of the proactive approach to address the issue of network fragmentation that occurs when the distance between vehicles exceed their transmission range [27, 34, 35].

Periodic approach. The periodic approach implements a store-and-forward module [29, 36, 37]. The messages are here periodically retransmitted and act as beacon messages. Cherif *et al.* introduce a protocol that makes use of the store-and-forward technique to handle intermittent connectivity [29]. Upon successful reception of a message, a vehicle stores the message so as to be able to farther disseminate it. When a next relaying node is found, the dissemination is delegated to the latter.

Another method consists in determining the node last connected to the current cluster which in turn, will be in charge of propagating the dissemination process to previously disconnected vehicles [36, 37]. A node that did not receive twice the same message or from is determined as the last connected and will take on the propagation of the dissemination by periodically rebroadcasting the message.

Reactive approach. This approach relies on the discovery of previously disconnected nodes. Upon discovering such node, the dissemination process is resumed [9, 27]. The farthest node in a cluster is the one left without receiving any acknowledgement when sending a message. This node will keep retransmitting the message until an upstream cluster of vehicles catches up from behind. The retransmission of the message is then stopped once acknowledged by a previously disconnected node. The dissemination process is resumed until the last connected vehicle of the upstream cluster is reached.

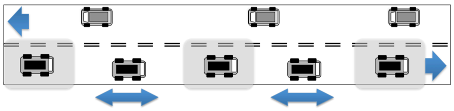


Figure B.2. Simulation scenario of the disconnection tolerance component.

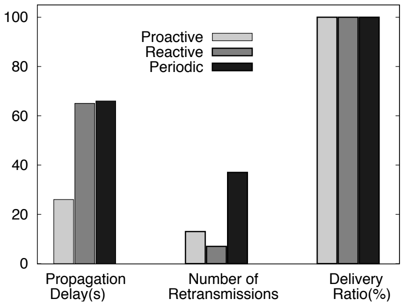


Figure B.3. Performance of the proactive, reactive, and periodic approaches of the disconnection tolerance component.

Evaluation. In our simulations, we analyze the performance of the three instances of the disconnection tolerance component in terms of propagational delay, number of retransmissions and delivery ratio. The latter metric refers to the number of vehicles that receive a message successfully to the total number of vehicles in the simulation. We simulate a scenario which consists of a 4,000 meter-long stretch of a long highway. Each vehicle has a transmission range of 250 meters and moves at speeds ranging from 46 km/h to 110 km/h for a duration of 100 seconds. In order to evaluate the performance of the different technical instantiations of the intermittent connectivity component, we consider a simulation scenario consisting of two types of nodes.

This simulation scenario is depicted in Figure B.2 where three *Constant Nodes* shown in a grey shaded box are moving toward the site of the incident, along with two *Connection Nodes*. Constant Nodes move at constant speed and thus remain at a distance from each other that exceeds their transmission range whereas Connection Nodes move back and forth between two consecutive Constant Nodes at different speeds preventing them from being neighbors of more than one Constant Node at the same time. The opposite traffic stream is also represented by means of three vehicles.

In Figure B.3, we can see that the proactive approach achieves the lowest message propagational delay. This result shows the benefit of using the vehicles moving in the opposite lane to speed up the dissemination process. However, this short delay comes at the expense of an increased number of broadcasts compared to the reactive approach since the proactive approach will send the report to vehicles driving along both lanes.

The reactive and periodic approaches show similar results in terms of propagational delay as both have to wait for a topology change within the flow of vehicles moving towards the accident in order to further continue the dissemination process. The main difference lies in the higher number of retransmissions required by the periodic approach to achieve the same delivery ratio in comparison to the two others approaches. This is due to the fact that when using the periodic approach, nodes will retransmit frequently as they have no way to learn about their new neighbors. On the contrary, the reactive approach will avoid unnecessary retransmissions unless previously disconnected nodes join the current cluster.

Takeaway B.8

The ability to disseminate information in vehicular networks is of paramount importance for a wide range of services and applications such as road safety or commercial advertising. An analysis of the existing dissemination protocols shows that most of them implement four basic conceptual components addressing the requirements of specific applications, as well as the challenges inherent to vehicular networks.

3.2 | *Servus: Low-cost reliable and disconnection-tolerant broadcasting in VANETs*

(Publication [9]: R. A. Gorcitz, P. Spathis, M. Dias de Amorim, R. Wakikawa, and S. Fdida. "Servus: Reliable low-cost and disconnection-aware broadcasting in VANETs." *IEEE 7th International Wireless Communications and Mobile Computing Conference (IWCMC*, 2011.)
(Ph.D. thesis of Raul A. Gorcitz, UPMC Sorbonne Universités, 2013 [10]. Co-supervised by Serge Fdida and Marcelo Dias de Amorim.)

In this section, we draw on the results of the analysis of the state of the art we conducted in previous section to propose the design of Servus, a robust protocol that guarantees data dissemination with high delivery ratio and low traffic overhead. Servus includes a new broadcast management mechanism that takes advantage of the inherent behavioral properties of the VANET environment.

In particular, Servus can update isolated nodes (or clusters) with missing information while ensuring homogeneous data dissemination at low overhead. Through a number of analyses, we show that Servus is highly efficient with regard to the tradeoff between reliability and traffic overhead by mitigating the broadcast storm problem and the topological temporal fragility. Servus brings four innovative contributions:

- **Dissemination relayer selection.** We propose a selection method for choosing the rebroadcaster that mitigates the broadcast storm problem. This method exploits the acknowledgment-based scheme introduced for reliable dissemination.
- **Opportunistic message overhearing.** We design a rumor mechanism that improves the dissemination delivery ratio by exploiting opportunistic message overhearing. This mechanism also limits message repetitions.
- **Connectivity gaps management.** We propose a proactive store-and-forward mechanism to handle connectivity gaps in the flow of vehicles. This mechanism enables the dissemination of missing messages inside clusters of vehicles previously disconnected.

- **Intersection management.** We include a novel method to manage intersection for urban scenarios. We use the vehicles that stop at the intersection as relays for disseminating in the other directions.

The evaluation results show that Servus achieves high delivery ratios in highway and urban scenarios, under various speeds and node densities. Servus also scales well and maintains a high delivery ratio with a minor increase in retransmissions. In Section 3.2.1, I describe the main functional components of Servus. The results of Servus performance evaluation are presented in Section 3.2.2.

3.2.1 Servus description

Servus ensures proper advertisement of an event by restricting the direction of its dissemination within a predefined area of interest. When a vehicle detects an abnormal event such as an accident or roadwork, the vehicle broadcasts a warning message advertising the coordinates of the event so all neighboring vehicles traveling in the direction of this event are informed. The warning message is then rebroadcasted in a multihop fashion so other vehicles are informed of the event. Servus mitigates the broadcast storm problem by selecting the furthest reachable vehicle as the only next forwarding vehicle. Those vehicles can be considered as a gateway for the vehicles located outside the initial transmission range.

Data structures. A vehicle maintains the following tables: (*i*) the Message Table and (*ii*) the Neighbor Table. The Message Table keeps track of all the messages received by a vehicle. The Neighbor Table contains the list of in-range neighboring vehicles who satisfy the *trajectory condition*: A neighboring vehicle is added to the Neighbor Table of a vehicle if the angle formed by their trajectories falls within a specific configurable interval.

Beaconing. Beaconing prevents Servus from relying on complex roadmaps. Vehicles periodically advertise their position and direction by sending a Hello message. A vehicle uses those neighbor information to compute the trajectory of the sending vehicle and checks the trajectory condition. If the condition is satisfied, the position of the neighbor is added to the Neighbor Table.

Multihop dissemination. Upon detecting an abnormal event, a vehicle broadcasts a warning message immediately. This message is only processed by the following vehicles driving in the direction of the event.

Upon receiving the warning message, the receiver vehicles check if the sending vehicle satisfies the trajectory condition. If yes, the vehicle is added to their Neighbor Table and the warning message is further processed. If the message is received for the first time, the receivers update their Message Tables with the information of the warning message. If the warning message is a duplicate or if the sender vehicle do not fulfill the trajectory condition, the message is discarded with no further actions.

Once the Message Table updated, a receiver vehicle returns an positive acknowledgement (ACK) confirming good reception of the warning message. On the sender side, the ACK cancels the message retransmission timer. ACKs are sent even in case of duplicated warning messages as those ACKs will refrain the sender vehicle from retransmitting the same message. As so, ACKs help decrease the traffic overhead on the network.

The receiver vehicles are then in charge of relaying the warning message to nearby following vehicles in the area of interest. Servus mitigates the broadcast storm problem by searching for the furthest receiver vehicle who will be in charge of rebroadcasting the message.

Reliability and opportunistic overhearing. After sending a warning message, a vehicle starts a transmission timer. If the vehicle receives an ACK for a message advertising the same event from a following vehicle, the receiver vehicle stops the retransmission timer and discards the warning message. If no such ACK is received before the timer expires, the vehicle rebroadcasts the message.

In addition to positive acknowledgments, Servus uses negative acknowledgments (NACKs) to ensure reliable transmission of the warning messages. A NACK message serves as a retransmission request that notifies a heading vehicle of a missing warning message. NACKs are generated following the opportunistic overhearing of an ACK.

Upon receiving a warning message, a vehicle returns an ACK that can be received by neighboring vehicles that have yet to be informed of the event. Those vehicles add the missing warning message as a 'rumor entry' in their Message Table and send a NACK message. Whereas ACKs are processed by preceding and following vehicles, a NACK is process by the only preceding vehicles. Rumor entries alleviate broadcast storm by preventing a vehicle from rebroadcasting a warning message in areas where an event has already been disseminated. The vehicles who sent the warning message in the first place processes a NACK by retransmitting the message while other receiver vehicles ignore the NACK.

Intersection Management. Thus far, we considered scenarios where warning messages are disseminated opposite to the driving direction toward a dangerous situation. Single-direction dissemination fits well with the highway scenarios. In Servus, we also consider urban scenarios where intersections are frequent. Vehicles near an intersection should disseminate the warning messages in the direction of each crossing road.

When a abnormal or dangerous situation is detected, a warning message is sent immediately and then rebroadcasted by the receiver vehicles in a multihop fashion so vehicles outside the transmission range of the initial sender are further informed of this situation.

In an intersection scenario, the vehicles near to the middle of an intersection play an important role in disseminating a warning message in the the many direction of the crossing roads. Vehicles traveling toward the event are therefore involved in the dissemination without the need of fulfilling the trajectory condition.

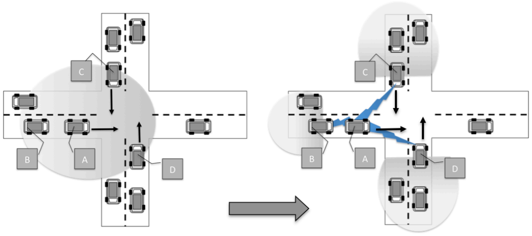
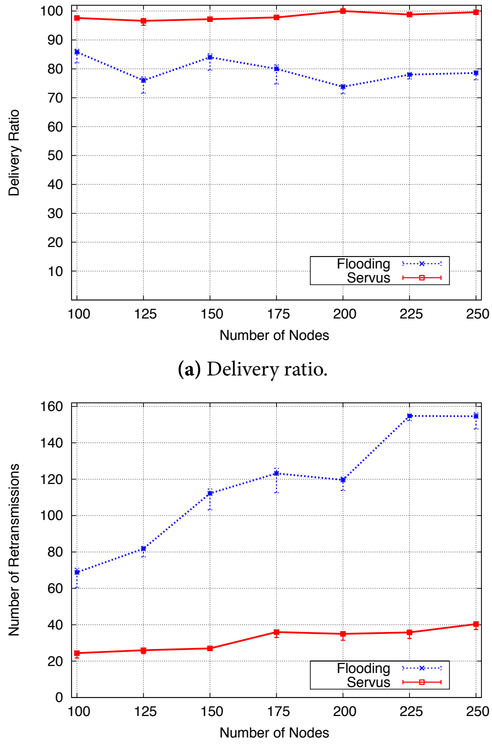


Figure B.4. Intersection management.

In Figure B.4, vehicle A detects a dangerous situation and broadcasts a warning message to the neighboring vehicles located in transmission range, including vehicles B, C, and D. C and D calculate the angle formed by the trajectory of A and their own trajectory. If this angle is lower than a predefined value (e.g. 90 degrees), the dissemination is propagated in the direction of the crossing streets by the vehicles traveling in the direction of the intersection. In Figure B.4, vehicles C and D rebroadcast the warning message respectively on the northbound and southbound roads. Vehicle B also participates to the dissemination of the message on the westbound road since it satisfies the trajectory condition with A.

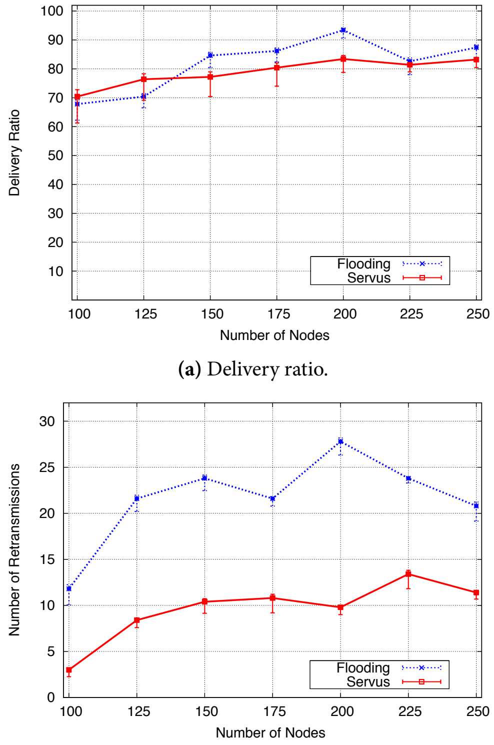
Disconnection tolerance. A vehicle relies on beaconing and neighbor information to determine if there are any following vehicle before rebroadcasting a warning message. If the vehicle is the furthest in the current cluster, this vehicle stores the messages in Message Table until a neighbor not listed in the Neighbor Table comes in retransmission range. The beacon messages received from this new neighbor triggers the update of the Neighbor Table and the transmission of the messages stored in the Message Table.



(a) Delivery ratio.

(b) Number of retransmissions.

Figure B.5. High scenario.



(a) Delivery ratio.

(b) Number of retransmissions.

Figure B.6. Urban scenario.

3.2.2 Evaluation

In this section, we evaluate the performance of Servus compared to flooding using NS-2 network simulator. We use two simulation scenarios including a highway scenario and urban scenario. We measure the delivery ratio, e.g., the number of vehicles informed of a dangerous situation to the total number of vehicles located in the area of interest, and the number of retransmissions required to achieve this ratio. In the following plots, the vertical lines indicate lower 95% confidence interval bounds.

Highway scenario. We simulate a 3,000×3,000 meter area where vehicles move in a straight line in both directions. The total number of nodes vary from 100 up to 250. We assume vehicles have a transmission range of 250 meters and the antennas omnidirectional. We use NS-2 with the TwoRayGround radio propagation model. Vehicle movement traces are generated with CanuMobiSim (CANU Mobility Simulation Environment) [3]. We use the graph-based mobility model [38] with random starting positions and varying speeds ranging from 50 to 110 km/h. In our simulations, one of the vehicles starts the dissemination by sending a 177-byte message.

In Figure B.5a, we can see that Servus scales well and achieves a delivery ratio close to 100% whereas the delivery ratio of flooding fluctuates around 80% as the node density increases. This result is due to the lack of disconnection management when flooding is used to disseminate warning messages. Once informed of a situation, the dissemination process stops at the last connected vehicles. New neighbor catching up later the cluster of informed vehicles are left out the dissemination process. In Figure B.5b, we measure the number of retransmissions generated by each approach to achieve these delivery ratio. We can see that flooding requires higher number of retransmissions as the number of vehicles increase. Warning messages are rebroadcasted even if requested by a single vehicle. In the case of Servus, the combination of ACKs and NACKs limits the number of unnecessary retransmissions.

Urban scenario. We simulate a 1,480×2,000-meter area of interest where vehicles move in both directions with speeds varying from 30 to 50 km/h. The other simulation settings are the same with the highway scenario. We use a grid road topology where a static vehicle located an intersection starts the dissemination of a warning message. Our simulation results are shown in Figure B.6.

In Figure B.6a, we can see that flooding achieves higher delivery ratios as the number of vehicles increases. Flooding turns all vehicles in rebroadcasters whereas Servus only selects strategically located vehicles with respect to the dissemination latency and the broadcast traffic overhead. The erratic variations observed for the delivery ratio achieved with flooding results from the connectivity gaps due to traffic dynamics. Nevertheless, Servus succeeds in achieving a 100% delivery ratio on the long run. As more vehicles traveling toward the intersection get closer, the delivery ratio increases once they arrive in the transmission range of the static vehicle or when they become neighbors with already informed vehicles.

The higher delivery ratios achieved with flooding come at the price of increased traffic overhead compared to Servus. As shown in Figure B.6b, flooding incurs a significant traffic overhead compared to Servus. As for the highway scenario, the higher number of retransmissions observed for flooding result of the many blind rebraodcasting rounds.

Takeaway B.9

Servus is a multihop broadcast-based protocol designed for reliable data dissemination in vehicular networks. Servus alleviates broadcast storms by selecting strategically located rebroadcaster vehicles without relying on road maps. Servus achieves high delivery ratios while limiting the number of retransmissions and successful dissemination over transient transmission gaps resulting from the traffic dynamics.

3.3 | Conclusion

In this chapter, I have presented a component-based analysis of the most relevant data dissemination protocols proposed in the context of VANETs. We identified a set of generic building blocks and compared the performance of a representative realization of each component with regard to their performance giving various applications profiles and environment parameters. This work helps better understand the impact of different requirements among existing and emerging new applications, as well as constraints due to the highly challenging environment of vehicular networks. Based on the results drawn from this analysis, we proposed the design of Servus, a broadcast-based dissemination protocol that efficiently tackles the challenges resulting from the dynamic structure of vehicular networks.

[3] http://canu.informatik.uni-stuttgart.de/mobisim/

Part C
Tapping into Everyday Mobility

In this second part of my document, I tackle mobility as an enabler with respect to data delivery. Instead of considering mobility as a challenge, I take benefit of the alternative communication channel resulting from the movements of entities such as vehicles equipped with storage capabilities. I use this alternative communication channel to complement or in replacement of conventional infrastructure-based data networks. Instead of considering the only forwarding opportunities resulting from the encounters between mobile entities, I study the benefits of the mobility phases that connect consecutive encounters. From my perspective, the existing mobility of surrounding entities can be exploited to overcome various limitations regarding the capacity and the coverage of conventional data networks.

The contributions presented in this second part are organized as follows:

Chapter C.1: Vehicular data offloading

> (Publication [39]: B. Baron, P. Spathis, H. Rivano, and M. Dias de Amorim. "Étude de l'intermodalité pour le délestage des réseaux d'infrastructure." *17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, 2015.)
> (Publication [40]: B. Baron, P. Spathis, H. Rivano, and M. Dias de Amorim. "Offloading massive data onto passenger vehicles: Topology simplification and traffic assignment." *IEEE/ACM Transactions on Networking*, 2015.)
> (Publication [41]: R. A. Gorcitz, Y. Jarma, P. Spathis, M. Dias de Amorim, R. Wakikawa, J. Whitbeck, V. Conan, and S. Fdida. "Vehicular Carriers for Big Data Transfers." *IEEE Vehicular Networking Conference (VNC)*, 2012.)
> (Publication [42]: S. Fdida, P. Spathis, and M. Dias de Amorim, "Massive Data Migration on Wheels." *IFIP/IEEE Wireless On-Demand Network Systems and Services (WONS)*, 2011.)
> (Submission : B. Baron, P. Spathis, H. Rivano, M. Dias de Amorim, Yannis Viniotis, and M. Ammar. "Centrally-Controlled Mass Data Offloading Using Vehicular Traffic." Submitted to *IEEE Transactions on Network and Service Management*, 2016.)
> (Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)
> (Ph.D. thesis of Raul A. Gorcitz, UPMC Sorbonne Universités, 2013 [10]. Co-supervised by Serge Fdida and Marcelo Dias de Amorim.)

In the first chapter of this part, I present a transmission system relying on the existing mobility of conventional vehicles. Data is stowed away on-board vehicles without their owners or drivers being aware. The latter drive the vehicles as they usually do in their normal routine. I study the flows of vehicles travelling their routine journeys as an alternative communication channel that network operators can use to extend cost-effectively their transport capacity. This work finds its motivation in the context of traffic offloading as a promising technique for alleviating the ever-growing traffic load from infrastructure-based networks such as the Internet.

Chapter C.2: Vehicular cloud services

> (Publication [44]: B. Baron, P. Spathis, M. Dias de Amorim, and M. Ammar. "Cloud Storage for Mobile Users Using Pre-Positioned Storage Facilities." *ACM MobiCom Workshop on Experiences with Design and Implementation of Smart Objects (SMARTOBJECTS)*, 2016.)
> (Publication [45]: B. Baron, M. Campista, P. Spathis, L. H. MK Costa, M. Dias de Amorim, O. Carlos MB Duarte, G. Pujolle, and Y. Viniotis. "Virtualizing vehicular node resources: Feasibility study of virtual machine migration." *Elsevier Vehicular Communications*, 2016.)
> (Submission: B. Baron, P. Spathis, M. Dias de Amorim, and Y. Viniotis. "Mobility as an alternative communication channel: A survey." Submitted to *IEEE CommunicationsSurveys & Tutorials*, 2016.)
> (Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)

In a second chapter, I study two cloud-like services which draw on the scheduled mobility of public buses to remove the dependency on conventional infrastructure-based data networks. I conclude this chapter with a survey of various mobility-assisted transmission systems proposed in the literature.

Vehicular file storage and sharing system: Bus stops are turned into repositories where mobile users upload their files from where the regular services of public buses are used opportunistically to replicate each file within every other repositories.

Virtual machine migration in vehicular cloud networks: Instead of relying on stationary nodes such as bus stops, I study the feasibility of transferring data between mobile entities in contact. I conduct this study in the context of a virtualized mobile network consisting of the public buses operated in the city of Dublin, Ireland. The reallocation of the virtual resources hosted requires the migration of the virtual machines hosted by the mobile nodes while in contact.

Mobility-assisted transmission systems: I review the literature that exploits the communication channel resulting from the movements of various entities equipped with storage capabilities. The focus of this survey is on the methods used to pass the data from one mobile entity to another.

Chapter C.1

Vehicular Data Offloading

In this chapter, I advocate the use of private vehicles equipped with storage devices with the aim of turning the road infrastructure into a large capacity transmission system. I propose to exploit the delay-tolerance of background traffic to offload bulk data transfers from an infrastructure-based network (e.g., the Internet) over the road network. Possible transfers include background data exchanged in the framework of applications with a delay tolerance ranging from several days to few weeks (e.g., distribution of large scientific datasets or data resulting from maintenance and provisioning activities).

I propose an vehicular offloading system that takes opportunistic advantage of the increasing number of journeys involving vehicles, as well as the time vehicles spend parked. According to recent reports, a vehicle spends 80% of its time parked at home or 17% at a destination and the rest of its time driving [46]. The idea is to stow away data on board of vehicles while parked, unbeknownst to their driver. The trips taken by vehicles loaded with data create an ancillary transmission medium which can bear large segments of traffic in support of the Internet. A back-of-the-envelope calculation shows that 10% of the vehicles traveling the roads of France equipped with a 1 TB hard drive can transport up to 115 EB per day (1.3 PB per second).

While this offloading system may look conceptually straightforward, the underlying paradigm raises as many challenges as avenues for research I followed for some and present in this chapter. The main step toward a vehicular offloading service can be summarized as follows: *Given a request to offload a data transfer, how to select the road paths and the flows of vehicles matching the data transfer requirements?* First, we need to cope with the high degree of complexity of the road network's topology and the large number of daily routine journeys involving vehicles. Second, we need a fair allocation process that maximizes the road traffic utilization.

The main contributions I present in this chapter are the following:

- **Vehicular offloading** (Section 1.1). I give an overview of the principles underlying the concept of vehicular offloading as well as the main components involved such as the offloading spots.
- **Centralized control of the road resources** (Section 1.2). I present an SDN-based architecture that enables efficient control over the road infrastructure to offload bulk delay-tolerant traffic from an infrastructure network.
- **Road map reduction** (Section 1.3). I present a reduction method that creates an abstract representation of the road infrastructure. The output of the reduction method is illustrated in the context of a network of charging stations for electric vehicles.
- **Vehicular allocation problem** (Section 1.4). We formulate the allocation of vehicular flows as a max-min fairness problem. This formulation maximizes the throughput of the data carried by vehicles while achieving fairness amongst competing transfers.
- **Real-world evaluation** (1.5). We evaluate the throughput of multiple offloading demands assigned onto the French road network using actual road traffic counts.

1.1 | *Vehicular offloading*

(Publication [39]: B. Baron, P. Spathis, H. Rivano, and M. Dias de Amorim. "Étude de l'intermodalité pour le délestage des réseaux d'infrastructure." *17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, 2015.)
(Publication [40]: B. Baron, P. Spathis, H. Rivano, and M. Dias de Amorim. "Offloading massive data onto passenger vehicles: Topology simplification and traffic assignment." *IEEE/ACM Transactions on Networking*, 2015.)
(Publication [47]: R. A. Gorcitz, Y. Jarma, P. Spathis, M. Dias de Amorim, R. Wakikawa, J. Whitbeck, V. Conan, and S. Fdida. "Vehicular Carriers for Big Data Transfers." *IEEE Vehicular Networking Conference (VNC)*, 2012.)
(Publication [42]: S. Fdida, P. Spathis, and M. Dias de Amorim, "Massive Data Migration on Wheels." *IFIP/IEEE Wireless On-Demand Network Systems and Services (WONS)*, 2011.)
(Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)
(Ph.D. thesis of Raul A. Gorcitz, UPMC Sorbonne Universités, 2013 [10]. Co-supervised by Serge Fdida and Marcelo Dias de Amorim.)

Private vehicles are equipped with one or more memory storage devices such as magnetic disks or other non-volatile solid-state storage devices. The term *vehicle* refers to both private vehicles and commercial vehicles. We assume that the content of the storage devices is not accessible to the drivers and the data encrypted when piggybacked onto the vehicle. Commercial vehicles may be part of a fleet of vehicles owned or leased by a business or a governmental agency. Vehicles also embed one or more communication network interfaces.

The flow of vehicles so equipped acts as a mechanical backbone connecting a collection of offloading spots. The term *offloading spot* refers to a fixed device offering short- to medium-term data storage, located where vehicles park for long enough as part of their line of travel. Examples of such locations include on-street parking spots, parking lots, or gas stations. In the case of electric vehicles, the offloading spots may be co-located with the battery charging stations.

Offloading spots serve two distinct purposes depending on their relative position with respect to the offloading process. The dual role of an offloading spot is depicted in Figure C.1, where a large amount of delay-tolerant background data needs to be transferred between two remote data centers.

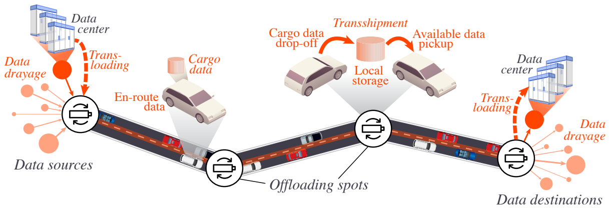


Figure C.1. Dual role of the offloading spots with respect to the process of offloading background traffic between two remote data centers onto the road network.

Part of or all the data originating from a conventional data network is first transloaded to the closest edge offloading spot until transferred onto passing vehicles. Subsequent offloading spots act as intermediate exchange relay points where data is transshipped from one vehicle to another: Vehicles can drop off their data cargo for later pickups by subsequent passing vehicles. Data is loaded on or off the vehicles using short-range radios, unbeknownst to their driver. Once at the destination, the data is transloaded back into the original data network. This vehicular offloading architecture can be seen as an intermodal transportation system which involves different modes of transportation for freight or passengers.

The offloading spots remove the need of relying on vehicles solely traveling all the way from the source to the destination of a data transfer. Every time a vehicle stops at an offloading spot, the direction of the vehicle is matched against the destination of the data transfers traversing the offloading spot. The vehicle unloads its cargo data if heading to a direction different from the destination of the corresponding transfer. The data is stored until transferred on a subsequent vehicle heading toward the intended destination. As a result, multiple consecutive offloading spots may be involved if the data needs to be shipped across a large body of country before reaching geographically long distant destinations. The data is 'hitchhiked' hop-by-hop through the network of offloading spots before reaching its final destination. A data transfer follows the same path as the flow of vehicles traveling the stretches of road connecting this sequence of offloading spots.

Takeaway C.1

The bandwidth resulting from the mobility of vehicles equipped with storage devices can help offload large segments of traffic in support of the Internet. Though the concept of vehicular offloading may seem straightforward, vehicular allocation raises many challenges including coping with the complexity of the road network and meeting the requirements of a data transfer in terms of capacity and delay.

In the next section, I present a SDN-like centralized architecture consisting of a controller and the collection of offloading spots acting as forwarding engines. The offloading spots take the decision whether to load data on or off vehicles according to forwarding states installed by the controller.

1.2 | *SDN-enabled road network management*

(Publication: [48]: B. Baron, P. Spathis, H. Rivano, M. Dias de Amorim, Y. Viniotis, and J. Clarke. "Software-sefined vehicular backhaul." *IFIP Wireless Days*, 2014.)
(Submission : B. Baron, P. Spathis, H. Rivano, M. Dias de Amorim, Yannis Viniotis, and M. Ammar. "Centrally-Controlled Mass Data Offloading Using Vehicular Traffic." Submitted to *IEEE Transactions on Network and Service Management*, 2016.)
(Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)

We leverage the advantages of the logical centralization provided by SDN (Software Defined Networking) to enable efficient control of the road infrastructure to offload bulk delay-tolerant data from an infrastructure network. SDN provides the logistics including planning, implementing, and controlling for the effective and efficient transportation of data over the road network.

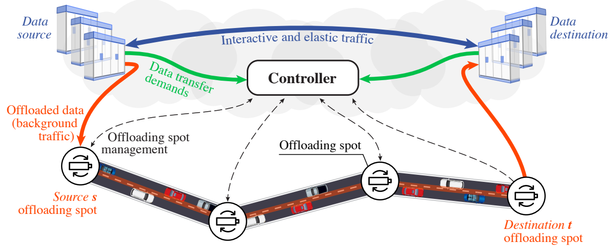

Data source
Interactive and elastic traffic
Data destination
Data transfer demands
Controller
Offloaded data (background traffic)
Offloading spot management
Offloading spot
Source *s* offloading spot
Destination *t* offloading spot

Figure C.2. SDN-like centralized control for the road network for offloading bulk transfers of delay-tolerant data.

Following SDN's original design, the centralized architecture I present in this section consists of two components, as depicted in Figure C.2: a central controller and the offloading spots acting as forwarding entities. The controller receives the demands to offload data transfers onto the road network. Each demand specifies the volume and delay requirements for the corresponding data transfer. The controller computes the road path consisting of a sequence of offloading spots connected by flows of empty vehicles whose number and speed match the data transfer requirements. The controller connects to the offloading spots and installs the forwarding states needed to select the vehicles that will carry the data to their final destination with respect to the transfer requirements.

In the SDN-like architecture I present in the next section, the controller maintains a holistic view of the road infrastructure and helps the offloading spots enforce an informed scheduling strategy for the efficient allocation of the road resources among competing data transfers.

1.2.1 Functional component-based breakdown

In this section, I describe the functions undertaken by each component of the SDN-like architecture for vehicular data offloading.

Controller. The controller maintains a holistic logical view of the offloading infrastructure, including the offloading spots and dynamics such as the traffic volumes on the road paths connecting the offloading spots. The controller may leverage traffic forecasting techniques or services such as Here[1], TomTom[2], or Airsage[3] to characterize the road paths in terms of bandwidth and to periodically update its logical view.

The controller uses this logical view to allocate the offloading demands and make efficient use of the road network resources. For reliability purposes, the controller keeps track of the progress of the data transfers at the offloading spots through a control channel (*e.g.*, a cellular network). Information about the data transfers includes the data waiting to be picked up at offloading spots and the data in transit.

The controller also receives statistics about the vehicles parking at offloading spots, including the historical locations of the vehicles made available via the navigation system. The historical locations are stored in a geographic database managed by the controller to help the offloading spots predict the remaining route of the parking vehicles. The controller can thus determine the next offloading spot the vehicles are more likely to visit on their route. Probabilistic tools, such as Hidden Markov Models [49], maximum entropy [50], or Bayesian networks [51, 52] can help infer the future trajectories of vehicles using the successive locations recorded by their route planner devices.

Offloading demands. The controller receives demands to offload part of or all the data belonging to a transfer on the road network. Each demand specifies the volume and delay requirements for the corresponding data transfer, as well as the data entry and exit points.

Upon receiving a request to offload a data transfer, the controller computes the road network paths by solving the data transfer allocation problem as a multi-commodity flow allocation model (presented in Section 1.4). A road network path consists of a sequence of offloading spots that can accommodate the data transfer requirements.

Solving the allocation problem also defines how much data to allocate to the flow of vehicles traveling the stretches of road connecting consecutive offloading spots along the road network path.

Offloading spots. Data is offloaded from a traditional data network to the closest offloading spot using a border dray transfer system. Offloading spots are featured with storage capabilities where data is stored until transferred to a parking vehicle via short-range radio.

As depicted in Figure C.3a, subsequent offloading spots act as data relay boxes where the data are dropped off for later pick-up and delivery by subsequent empty vehicles. The decision to load data on or off a vehicle are dictated by the controller and results from matching the direction of the vehicle against the destination of the transfer the data belongs to.

As so, the offloading spots act as forwarding engines that select empty vehicles based on their destination to move the offloaded data toward their final destination. Vehicle selection is also driven by the efficient use of the road network resources shared among competing data transfers.

Flow tables. The flow tables determine the forwarding behavior of an offloading spot. They match the direction of a passing vehicle and the destination of the data available locally. A flow table consists of a list of entries, each installed for an individual data transfer.

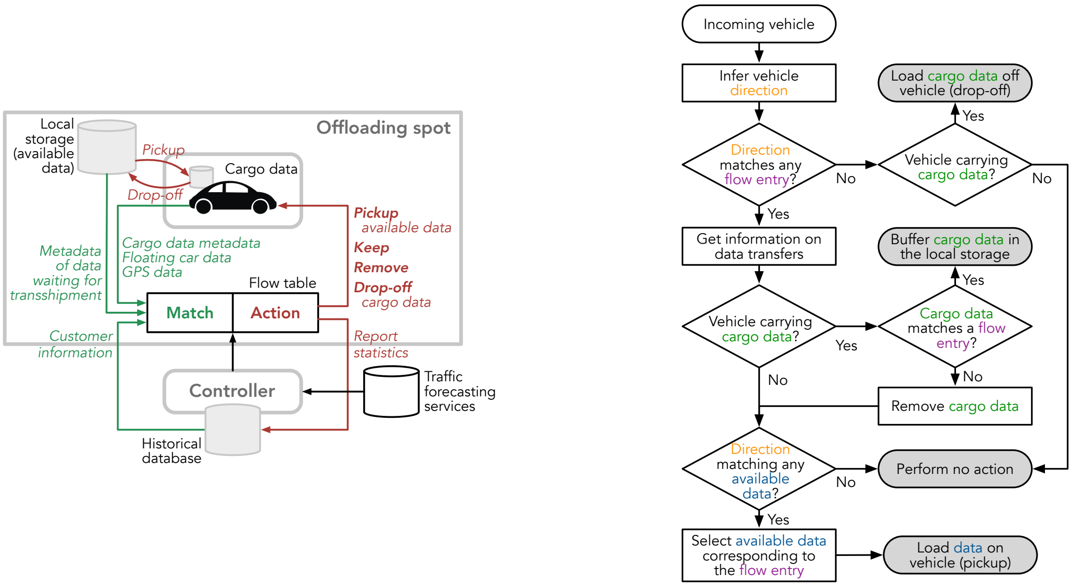
The controller adds a new entry in the flow table of the offloading spots located on the road network path computed for a data transfer. A flow table entry contains the next-hop offloading spot to which the data must be forwarded to reach the destination of the data transfer corresponding to this entry.

As depicted in Figure C.3a, a flow table entry also contains a list of actions to perform on the data. Common actions include loading data on or off the vehicles while parking close to the offloading spot. The controller defines these actions based on the information the offloading spots report on the flows of vehicles.

[1] https://www.here.com/business/traffic

[2] http://automotive.tomtom.com/en/connected-services/tomtom-traffic

[3] http://www.airsage.com/Products/Traffic-Insights/


Offloading spot
Local storage (available data)
Pickup
Drop-off
Cargo data
Metadata of data waiting for transshipment
Cargo data metadata
Floating car data
GPS data
Flow table
Match
Action
Pickup available data
Keep
Remove
Drop-off cargo data
Customer information
Report statistics
Controller
Traffic forecasting services
Historical database
Incoming vehicle
Infer vehicle direction
Direction matches any flow entry?
No
Yes
Vehicle carrying cargo data?
Yes
No
Load cargo data off vehicle (drop-off)
Get information on data transfers
Buffer cargo data in the local storage
Vehicle carrying cargo data?
Yes
No
Cargo data matches a flow entry?
Yes
No
Remove cargo data
Direction matching any available data?
No
Yes
Perform no action
Select available data corresponding to the flow entry
Load data on vehicle (pickup)

(a) For each transfer passing by an offloading spot, the controller installs a forwarding state including the list of actions to perform upon the arrival of a vehicle. Each action defines the behavior that the offloading spot needs to perform with the data belonging to the corresponding transfer.

(b) The offloading spot makes the forwarding decision by looking up the destination of a passing vehicle in its flow table. A matching flow table entry specifies the actions to perform with the data belonging to the corresponding transfer. The data can either be already stored at the offloading spot or carried by a passing vehicle.

Figure C.3. Forwarding at the offloading spot.

Forwarding process. The forwarding process determines the data to load on or off the vehicles parking at the offloading spots. Figure C.3b gives a flowchart representation of the forwarding process.

Upon the arrival of a vehicle, an offloading spot checks if the travel direction of the vehicle matches one of its flow table entry. If no matching entry is found, the vehicle unloads, if any, its cargo data into the offloading spot storage for future pick-ups and continues its journey without performing any further actions.

If multiple entries match the direction of an empty vehicle, the offloading spot selects the entry so as to ensure fairness among competing data transfers. After selecting one of the entries, the offloading spot performs the actions specified in the entry. If the vehicle already carries data, the offloading spot checks if this data belongs to the data transfer corresponding to the matching entry. If yes, the vehicle keeps its cargo and continues its journey. A local copy of the data is stored for later retransmissions in case the vehicle fails in delivering its cargo.

Otherwise, the vehicle unloads its cargo at the offloading spots before resuming its journey. In case the vehicle arrives empty, the offloading spot checks if some data matching the vehicle direction was locally dropped by a previous vehicle or transshipped from the infrastructure-based network. If such data is found, the data is loaded on the vehicle. Otherwise, the vehicle continues its journey empty-loaded.

1.2.2 Reliability

The controller uses retransmission and redundancy mechanisms to mitigate the effects of data losses resulting of vehicles unexpectedly changing direction, vehicles hijacking, accidents, or breaks down. These events are captured by the parameter we refer to as the *data leakage*.

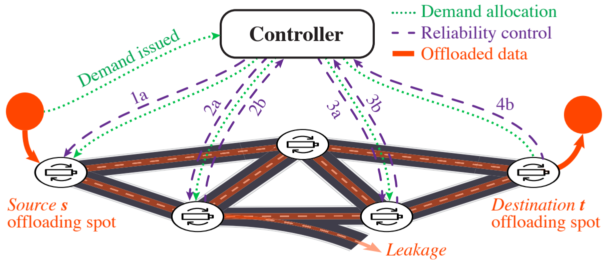


Figure C.4. Controller-to-offloading spot communication for the *hop-by-hop* (dashed arrows 1a-3a and 2b-4b) and *end-to-end* (dashed arrows 1a, 4b) retransmission strategies.

Retransmissions. Retransmissions of the cargo data completely recover from vehicles failing to deliver the data cargo to the intended offloading spots. We propose two retransmission strategies, as depicted in Figure C.4: hop-by-hop (hbh) retransmissions and end-to-end (e2e) retransmissions.

With the hop-by-hop strategy, each offloading spot buffers data for later recovery in case a vehicle fails to deliver its cargo to the next-hop offloading spot. The controller receives acknowledgments over a feedback channel from the next-hop offloading spot (indicated by dashed arrows 2b-4b in Figure C.4) and notifies through the same channel the previous offloading spot when to retransmit the missing data.

With the end-to-end strategy (indicated by dashed arrows 1a and 4b in Figure C.4), the only offloading spots in charge of recovering a loss are the edge offloading spots where the data is first transloaded from the computer network and stored until successfully transmitted. After a predefined number of attempts, a loss is repaired via the original computer network (from where data was first transloaded) to make sure the deadline specified in the offloading demand can be met.

Redundancy. To limit the number of retransmissions, we also use redundancy mechanisms, such as RAID. Redundancy reduces the effect of data leakage by transferring redundant copies of the data, in addition to the original data. The number of redundant copies is determined by the redundancy mechanism (*e.g.*, for one piece of data, two copies are needed with RAID 1, while RAID 6 adds two redundant copies to an array of n data, $n - 2$ being available before redundancy).

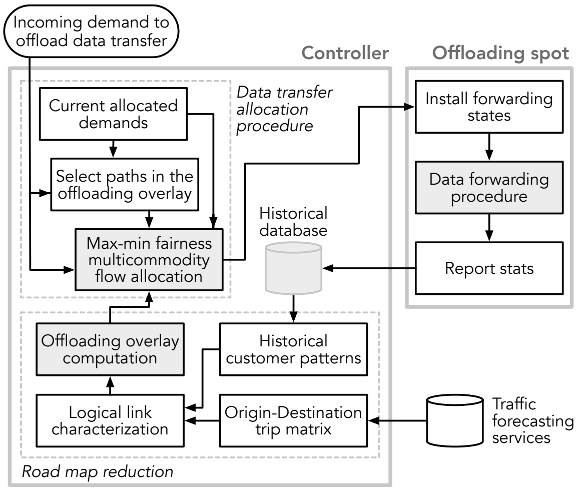
Takeaway C.2

Offloading spots can be seen as forwarding engines as long as equipped with a forwarding flow table. Such a table holds a set of flow entries, each of which contains the list of actions to perform on the passing vehicles depending on their direction of travel. Common actions include loading data on or off the vehicles. A SDN-like controller creates and installs those tables so as to match the performance requirements of the offloading demands.

In the following sections, I complete the functions of the controller by describing the steps taken to allocate the flows of vehicles depending on the performance requirements provided in the offloading demands. A chart of the interactions between the functions of the controller with those of the offloading spots is provided in Figure C.5.

The controller creates a logical representation resulting from a map reduction method presented in Section 1.3. This representation termed *offloading overlay* mitigates the complexity of the road network topology as well as its traffic volumes. The output of the reduction method is illustrated in the context of a networking of charging stations we use as a reference scenario in our evaluation.

The controller receives the demands to offload data transfers. The controller selects the flows of vehicle that match the performance requirements of each demand by solving the vehicular allocation problem presented in Section 1.4. We model the vehicular allocation as a max-min fairness problem for efficient and fair utilization of the road resources. The offloading overlay allows the use of linear programming techniques to solve the, otherwise intractable, problem of allocating vehicles to offload data transfers.



Incoming demand to offload data transfer
Controller
Offloading spot
Current allocated demands
Data transfer allocation procedure
Install forwarding states
Select paths in the offloading overlay
Data forwarding procedure
Historical database
Max-min fairness multicommodity flow allocation
Report stats
Offloading overlay computation
Historical customer patterns
Logical link characterization
Origin-Destination trip matrix
Traffic forecasting services
Road map reduction

Figure C.5. Interactions between the functions of the controller and those of the offloading spots.

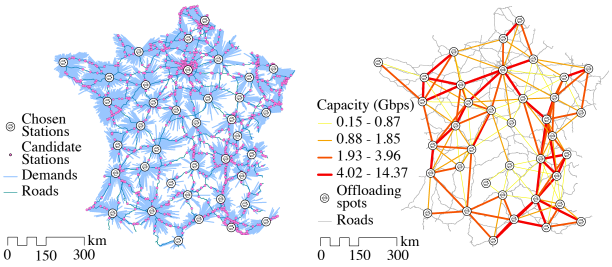
1.3 | *Road map reduction*

In this section, I present a reduction method inspired from transportation planning that creates a logical representation termed offloading overlay. In this representation, nodes correspond to the offloading spots connected by logical links. A logical link corresponds to the flows of vehicles traveling the road segments connecting two adjacent offloading spots in the road network. The objective of the offloading overlay is to mitigate the complexity of the road network.

Our reduction method takes as input the location of offloading spots. We illustrate the output of this method in the context of a network of charging stations. The data is carried by electric vehicles and the role of the offloading spots is undertaken by the battery charging stations.

In the following, I first present the dataset used for the deployment of a realistic charging stations placement for France. I then present the offloading overlay resulting of the reduction method. The logical links and paths in the offloading overlay are characterized by a set of attributes including the capacity, the travel time, and the data leakage.

Dataset. Our deployment framework and reduction method exploit the AADT (Annual Average Daily Traffic) made available to the public for most road networks in Europe, the Americas, and Asia. The AADT is the total volume of traffic passing a stretch of road in both directions for one year, divided by the number of days in the year. The traffic volumes are collected using strategically located automatic traffic recorders. The AADT is a fundamental statistic used in traffic engineering. The use of the AADTs helps reduce the effects of seasonal bias and missing data mainly due to equipment failure, construction schedules, and installation dates that plague continuous traffic monitoring. Traffic volumes for each road segment can be directly obtained by multiplying AADT by the duration in days of a transfer.



(a) Deployment of a charging station network and road map reduction for France.

(b) Offloading overlay with the average capacity of each logical link.

Figure C.6. Facility-allocation result and offloading overlay. The big dots are the chosen stations.

Charging stations deployment. The charging station deployment framework we propose takes in addition to the road map data for France, two sets as inputs: a set of candidate locations consisting of the 1,024 gas stations operated by Total, a major oil company in France, and a set of demand points consisting of the 9,555 cities of France with a population of more than 1,000. The framework performs optimal placement by minimizing the number of existing gas stations where to open a charging station while maximizing the satisfied demands within a range of 150 km. The charging station placement is formulated and solved as a facility location problem we adapt from the maximal covering location problem [53].

The output of the deployment framework consists of 38 charging stations scattered on the French road infrastructure, as shown in Figure C.6a. We note that stations are mainly allocated near major cities, as the demands from urban areas are higher compared to rural areas.

Offloading overlay characterization. To characterize the capacity of the logical links in the offloading overlay, we need the origin-destination matrix between the offloading spots. However, the AADT cannot be used as is. The AADT gives the traffic counts as the number of vehicles traveling a stretch of road per unit of time. Therefore, we propose an algorithm that borrows traffic forecasting techniques from transportation research to estimate the origin-destination matrix of the offloading spots from the traffic counts. The algorithm consists of the following steps:

1. *Route determination.* The first step selects a subset of the many alternative routes connecting each pair of adjacent offloading spots in the road network. The selection consists in choosing the k-shortest routes in terms of travel time. The routes are also selected such that they share a low degree of similarity in terms of stretches of road in common. We implement this selection process by using the algorithms proposed by Abraham *et al.* [54].
2. *Route assignment.* The second step consists in assigning weights to the selected routes using the C-logit route assignment model [55]. The value of a weight is determined according to properties such as the travel time and the distance of the route. Those weights reflect the capacity of a route in attracting traffic, the higher the weight of a route the more traffic it will receive. The weights are then used in combination with the traffic counts to estimate the traffic volume of the routes selected in the first step between each pair of adjacent offloading spots.
3. *Trip matrix estimation.* In the third step, we use the entropy maximization model proposed by Zuylen and Willumsen to compute the origin-destination trip matrix consisting of all pairs of offloading spots in the offloading overlay [56]. This model determines the most likely distribution of the traffic across all the routes selected in Step 1 subjected to two constraints, namely the traffic counts of the routes' stretches of road and the C-logit weights.
4. *Logical link characterization.* Finally, we characterize each logical link (i, j) in the offloading overlay using the following attributes:
   - *Capacity* $c(i, j)$. The capacity of (i, j) represents the combined storage of all vehicles travelling between i and j. The capacity also reflects the market penetration ratio, i.e. the ratio of vehicles equipped with data storage devices.
   - *Travel time* $t(i, j)$. The transit time is computed as the travel time average for each route selected in the first step between i and j and weighted by the route weights computed in the second step.
   - *Leakage* $l(i, j)$. The leakage represents the ratio of vehicles that fail to deliver the data they transport to the next offloading spot.

Takeaway C.3

Techniques borrowed from transportation planning help reduce the complexity of the road network. The resulting simplified representation makes linear programming applicable to solve the vehicular allocation problem. Real-world traffic volumes are translated in networking quantities such as capacity and delay for the roads connecting a collection of battery charging stations deployed in France.

In the rest of this chapter, we assume that the capacity of the offloading spots is not limited and that there are no constraints on the number of transfers they can serve. An offloading demand d_{st} represents a request to offload a data transfer between source s and destination t. The demand is characterized by the amount of data β_{st} and the deadline τ_{st} before which the transfer should be completed.

For simplicity, we model the rate of the demands at s by a Poisson distribution λ_{st} and its mean value is the average throughput β_{st}/τ_{st}. We denote by $\mathcal{P}_{st}$ the set of all possible logical paths between the source and destination, respectively s and t. Each logical path $p \in \mathcal{P}_{st}$ consists of a sequence of logical links connecting adjacent offloading spots in the offloading overlay.

1.4 | *Vehicular allocation problem*

The controller receives the demands to offload data transfers. A demand is provided with the performance requirements for the data transfer. The task of the controller consists in selecting the empty vehicles traveling in the direction of the intended destinations such that (*i*) the number of vehicles is sufficient to meet the transfer requirements and (*ii*) the allocation of the vehicles' combined storage is efficient and fair among the competing transfers.

To this end, the controller starts by computing the logical paths consisting of a sequence of logical links selected according to their properties as specified in the offloading overlay. The controller configures then the offloading spots along the selected logical paths. The computation of the logical paths follows a dynamic allocation procedure.

In the following, $\mathcal{P}_{st}$ denotes the set of candidate logical paths between s and t. Each logical path $p \in \mathcal{P}_{st}$ consists of a sequence of logical links connecting pairs of offloading spots in the offloading overlay. The travel time $t(p)$ experienced by a data transfer allocated to logical path p is determined by the sum of two components: the transit component and the waiting component. The transit component is the sum of the transit time of each logical link in p. The waiting component is the sum of the waiting times experienced at each offloading spot connecting those logical links. We express $t(p)$ as a function of $R(i, j)$, the average number of transmissions of a chunk of data on logical link $(i, j) \in p$:

$$t\,(p) = \sum_{(i,\,j) \in p} R(i,\,j)\left[\delta_i + t(i,\,j)\right], \tag{1.4.1}$$

where δ_i is the waiting time at offloading spot i.

Linear programming formulation. We formulate the vehicular allocation problem as a linear programming (LP) model that determines the logical paths matching the performance requirements of the offloading demands. The LP consists in allocating $f(p)$ flows of data on the vehicles travelling the logical paths listed in $\mathcal{P}_{st}$. We first present the inputs and then the procedure we use to solve the vehicular allocation problem. This procedure relies on a multi-commodity flow allocation problem we formulate as a linear programming model.

Inputs. The procedure that allocates the data transfers takes as input the set $\mathcal{D}$ of all offloading demands on the road network. This set includes the previous demands already allocated in addition to the new demands. The allocation procedure also takes as input $\mathcal{P}_{st}$, the set of candidate logical paths between each pair s and t for all demands in $\mathcal{D}$. To enumerate the logical paths in $\mathcal{P}_{st}$, we propose to use Yen's k-shortest paths algorithm or a breadth-first search algorithm. In our simulations, we reduce the search space by considering the logical paths sorted according to the transit time of a single cargo data along each path. The offloading overlay and the properties of each logical link (*e.g.*, capacity, travel time, and data leakage) are also inputs of the allocation procedure.

Procedure. For each demand in $\mathcal{D}$, the controller allocates $f(p)$ flows of data to the logical paths of $\mathcal{P}_{st}$ according to the Max-Min fairness strategy. The Max-Min fairness strategy proceeds by successive iterations. The first iteration allocates the minimum flows to satisfy the requirements of the demands. The following iterations successively allocate the remaining capacity of the network to the demands that can receive more flow. More specifically, iteration i maximizes the minimal flow allocation noted ϕ_i and fixes the allocation for the demands that cannot be better served, i.e. because of the capacity constraints of the paths or if the demand requirements are already satisfied. The following iterations process the remaining demands. To determine for which transfers the current allocation can be further increased in the following iterations, we use a non-blocking test.

The non-blocking test is derived from the algorithm proposed by Pióro *et al.* [57]. This test compares the maximal throughput of the flows allocated by a multi-commodity flow allocation to the one resulting from the minimal flow allocation ϕ_i. If the multi-commodity flow allocation improves the maximal amount of data allocated to a demand, the demand will be fixed in the next iterations of the max-min fairness strategy. Otherwise, the demand cannot be better increased, and it is fixed to ϕ_i.

1.4.1 Data scheduling

Multiple entries in the flow table of an offloading spot i may match the direction of a vehicle. Each entry corresponds to different data transfers or the same transfer spanning many paths in the road network. The offloading spot selects in which order the flows of vehicles should be allocated to the transfers according to a scheduling policy configured by the controller. We denote by $f(d_k p_l)$ the flow of vehicles travelling logical path p_l allocated to the transfer resulting of offloading demand d_k. The controller assigns a weight $w(d_k p_l)$ to the transfers allocated to the logical paths in $P = \{d_k p_l | \forall d_k \in D,\, p_l \in \mathcal{P}^{d_k},\, (i,\, j) \in p_l\}$. P is the set of logical paths passing by offloading spot i and sharing the same next-hop offloading spot j. The controller computes $w(d_k p_l)$ by normalizing the rate of flow $f(d_k p_l)$ with the total rate of the flows travelling all paths in P:

$$w(d_k p_l) = \frac{f(d_k p_l)}{\sum_{p \in P} f(p)}. \tag{1.4.2}$$

The weights are used with a scheduling algorithm to determine in which order to assign data transfers to a passing vehicle if multiple transfers traverse the same offloading spot. In our simulations, we considered a weighted round robin scheduler [58].

Takeaway C.4

The controller receives the offloading demands and solves the vehicular allocation problem as a multi-commodity flow problem. The controller has a simplified representation of the road network resulting of techniques borrowed from transportation planning. The resulting allocation consists of a set of travel paths and the flows of vehicles travelling those paths allocated among the offloading demands with fairness.

1.5 | *Evaluation on the French road network*

In this section, I first present the simulation settings and then, the evaluation results. The objective of the simulation is to evaluate two metrics: (*i*) the delay to transfer pre-defined amounts of data depending on the number of offloading spots involved in the data transfers and (*ii*) the fairness of the allocation of concurrent transfers when using logical paths with similar lengths.

We consider the data transfer allocation procedure in the context of a network of charging stations for electric vehicles deployed across France. We evaluate the performance resulting from the allocation of data transfers across France in terms of maximum throughput and end-to-end delay. In the rest of this section, we consider a conservative market penetration ratio of 10%. The market penetration ratio represents the share of vehicles equipped with storage capabilities and ready to participate in our offloading service.

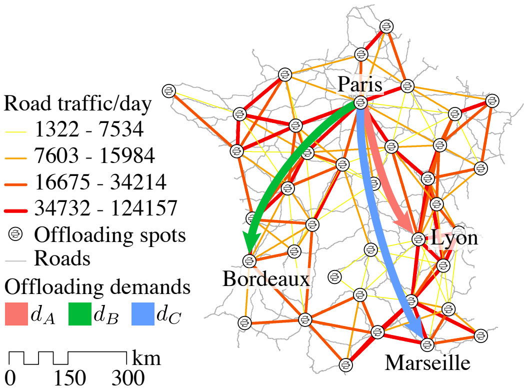


Figure C.7. Offloading overlay representing the network of charging stations for electric vehicles deployed to cover the roads of France. The thickness of the lines are proportional to the computed capacity of the logical links.

Dataset. In our evaluation, we use a dataset collected in 2011 featuring the AADT (Annual Average Daily Traffic) of the major roads in France covering a combined distance of 20,000 km [59].

We use the deployment framework presented in Section 1.3 to create a realistic deployment of charging stations covering all of France as depicted in Figure C.7. The charging stations are located 150 km apart, and their placement is determined by solving the facility allocation problem [60]. The resulting network of charging stations helps extend the driving range of the electric vehicles, while minimizing the number of charging stations.

Traffic origin-destination matrix estimation. We connect neighboring charging stations via a set of disjoint alternative routes selected in the road map of France by running the algorithms proposed by Abraham *et al.* [54]. Selected routes share up to 80% with the shortest route, while their length is at least 80% of the shortest route. To estimate traffic volumes, we use the C-logit traffic assignment model [55]. This model assigns a weight to the routes connecting the pairs of offloading spots located within a radius of 300 km (i.e. the driving range of an electric vehicle). We use the entropy maximization model proposed by Zuylen and Willumsen [56] to infer the origin-destination traffic matrix consisting of all pairs of offloading spots. We feed this model with the actual traffic counts provided by the AADT (Annual Average Daily Traffic) of the major roads in France covering a combined distance of 20,000 km[4].

Data transfer allocation. We evaluate the performance of the transfers resulting from the allocation of three offloading demands on top the offloading network consisting of charging stations deployed in France as described above. The three demands are shown in Figure C.7:

- d_A from Paris to Lyon with arrival rate λ_A,
- d_B from Paris to Bordeaux with arrival rate λ_B, and
- d_C from Paris to Marseille with arrival rate λ_C.

Note that the road paths followed by the transfers resulting from demands d_A and d_C share the same logical links in the offloading network; as so d_A and d_C are competing over those links. The duration of each simulation run lasts 300,000 seconds (3.5 days), which includes 43,200 seconds (12 hours) of *warmup*, to give time for the first data cargo to reach their destination.

[4] Census of the road traffic on the French road network in 2011 (in French): http://tinyurl.com/otfbewv

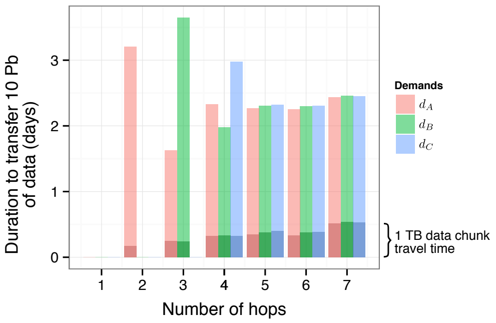


Figure C.8. The duration needed to complete a 10 Pb transfer (in lighter colors) and the travel time of a 1 Tb data cargo (in darker colors) as a function of the candidate path maximal length in terms of hops.

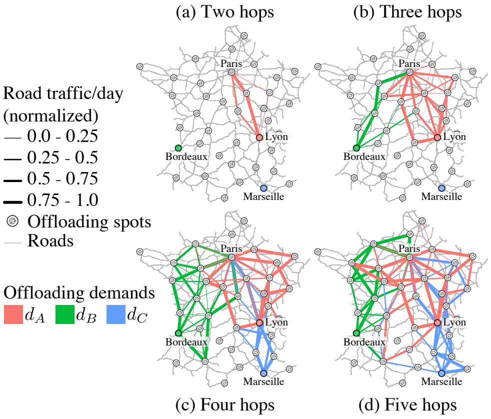


Figure C.9. Representation of the allocation of demands d_A, d_B, and d_C with different values for the maximal length of the candidate logical paths.

1.5.1 Number of offloading spots per data transfers

We first examine the impact of the number of offloading spots on the duration needed to complete demands d_A, d_B, and d_C. We consider that each demand is concerned with a transfer of 10 PB of data. Data losses are recovered by using the hop-by-hop strategy given that all logical links share a data leakage of 30%. The results are shown by the bar plot in Figure C.8.

We measure the transfer duration for d_A, d_B, and d_C as a function of the maximal length of the logical paths followed by each transfer expressed in terms of number of offloading spots. We also measure the mean travel time of a 1 TB cargo which corresponds to the cargo size of a vehicle. Our objective is to show the fairness in the allocation of the transfers as a function of the degree of similarity of the paths they follow. We examine the results in Figure C.8 together with Figure C.9 where we represent the logical paths allocated for each demand depending on the maximal length of the candidate paths.

We observe that none of the three destinations can be reached with a one-hop logical path. By increasing the logical path maximal length up to two hops, Lyon becomes the only city that can be reached, as shown in Figure C.9(a). The high duration for d_A is due to the low number of paths available and therefore of allocable vehicles, which results in a low throughput.

If we consider logical paths of three hops or less, Bordeaux is now reachable in addition to Lyon. Figure C.9(b) shows that, in addition to the two-hop paths, there are more candidate paths between Paris and Lyon. As a result, more vehicles are allocated to d_A which decreases its transfer duration. Regarding transfer d_B, the long transfer duration is explained by the few logical three-hop paths connecting Paris to Bordeaux in a similar way to d_A and the logical paths of two-hop maximum length.

With four-hop logical paths, Marseille is now also reachable, as shown in Figure C.9(c). Nevertheless, the number of four-hop logical paths is still limited between Paris and Marseille, in a similar way as the two-hop paths to Lyon and the three-hop paths to Bordeaux. What is more, Marseille is located farther away from Paris compared to Lyon and d_C competes for logical paths already passing by Lyon. This results in a longer transfer duration for d_C but also in an increase of d_A transfer duration. At the same time, increasing the length of the candidate paths to four hops enables the allocation of more logical paths between Paris and Bordeaux, which results in a clear decrease in the transfer duration of d_B. This decrease is also explained by the low degree of similarity between the logical paths allocated to d_B and those allocated to d_A and d_C, as shown in Figure C.9(c).

Finally, with logical paths of five hops and more, the transfer durations are equivalent among all the demands. This further confirms that the vehicle flow allocation is fair in terms of throughput among all the demands. A slight increase in the transfer duration for all demands follows each increment in the number of hops as a direct consequence of the longer logical paths followed by all transfers. A similar trend can be observed for the travel time of 1 TB cargo.

1.5.2 Cumulative capacity

We increase the stress on our system by allocating ten concurrent demands, all issued from Paris to the top nine other cities in France and to Basel located in Switzerland. This is shown in Figure C.10. We allocate the 10 demands at the same time and consider again the hop-by-hop retransmission mechanism, given a 30% data leakage for all logical links. We compute the cumulative throughput achieved by all ten transfers as a function of the length, expressed as the number

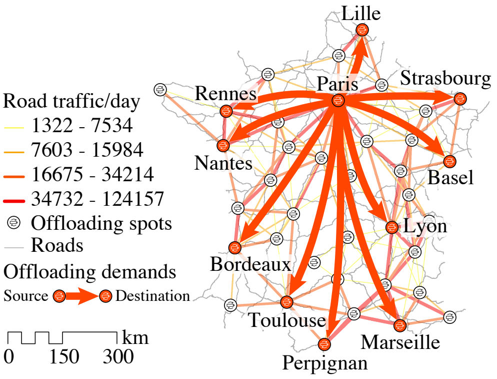


Figure C.10. Offloading demands to be allocated between Paris and the top 10 other major cities of France, including Basel in Switzerland.

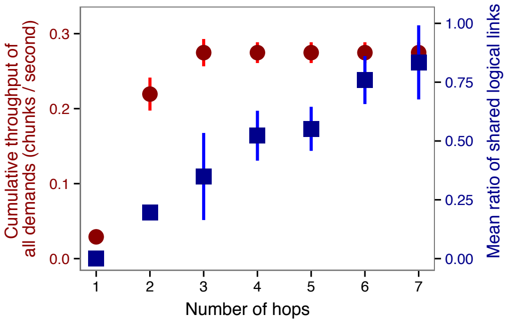


Figure C.11. The cumulative throughput of the allocation of the 10 competing demands shown in Figure C.10 and the mean ratio of shared logical links between the logical paths allocated to each demand over the ones allocated to all other demands.

of offloading spots of the logical paths allocated. We also compute the mean ratio of logical links shared by the logical paths allocated to each demand over the ones allocated to all other demands. The results are shown in Figure C.11.

As for the previous case regarding demands d_A, d_B, and d_C, all 10 destinations can be reached from Paris through logical paths of at least three hops. With at least three-hop long logical links, the cumulative throughput of all 10 demands reaches 280 Gbps for 1 TB chunks and remains the same with a low standard deviation among all demands. The only city reachable by one-hop logical paths is Lille while Lyon and Lille are the only cities reachable with at least two-hop paths.

As shown in Figure C.10, Lille is the closest to Paris compared to all other nine cities and can be reached by logical paths which exhibit the lowest degree of similarity with the paths connecting Paris to the other cities. For this reason, the transfer to Lille achieves a higher throughput compared to all other transfers.

For the destinations located at least three hops away from Paris, the cumulative throughput remains the same even after increasing the number of hops of the allocable logical paths. Extending the length of the acceptable paths should allow the allocation of a higher number of paths to each demand and thus increase the cumulative throughput. However, most of the newly added paths are too long to be allocated or are already used for other transfers toward closer cities.

For this reason, considering more paths by relaxing the length limit brings no benefit in terms of throughput. Considering longer paths to be allocable results in increased durations to complete the transfers. This further confirms the observation we made for Figure C.8, i.e. our vehicular allocation achieves fairness among all competing transfers.

Takeaway C.5

The evaluation results obtained with real-world traffic counts for France show that our procedure allocates the flow of vehicles by achieving fairness amongst competing transfers. We also showed that the benefits brought by allocating vehicles travelling longer travel paths is limited with regard to the throughput of the offloaded transfers while postponing the transfer completion time.

1.6 | Conclusion

In this chapter, I introduced the concept of vehicular data offloading. This concept exploits the existing mobility of private vehicles equipped with storage devices to offload large amounts of delay tolerant traffic from convention data networks such as the Internet. The flows of vehicles so equipped act as a mechanical backbone connecting pre-positioned data storage devices termed offloading spots, where vehicles usually park as part of their line of travel.

To assess the capacity enhancement brought by offloading data on vehicles, We formulate the allocation of vehicle flows as a max-min fairness multicommodity problem that maximizes the utilization of the flows of vehicle while ensuring fair allocation of these flows among concurrent data transfers.

To solve the vehicle flow allocation in reasonable computational time, I presented a road map reduction method that mitigates the complexity of the road network. Techniques from transportation planning help translate the attributes of a road traffic into networking quantities such as capacity and delay. This method also involves a deployment framework for optimal offloading spot placement.

To enable efficient control over the vehicular offloading resources, I propose a scalable architecture consisting of a SDN-like central controller that configures the offloading spots as forwarding engines. The controller has a logical view of the road network resulting of the road map reduction method and solves the max-min fairness multicommodity flow problem. The numerical results using real world traffic counts for France confirm the efficiency and fairness of our vehicle flow allocation.

Chapter C.2

Vehicular Cloud Services

In this chapter, I present two extensions I made to the work I conducted in the context of traffic offloading. The essence of those extensions came by considering the offloading spots from a new perspective. The offloading spots can be seen as locations where segments of trajectories made by independent mobile nodes can be concatenated in a single path followed by the data stowed away onboard the nodes. The storing capabilities of offloading spots allow the data to be passed asynchronously from one mobile node to another. In the following, the concept of offloading spot is extended according to two distinct directions I present in this chapter organized as follows.

- **Vehicular file storage and sharing system** (Section 2.1). The storage capacity of the offloading spots are used for the design of vehicular-based cloud system for storing and sharing files. The offloading spots act as repositories where files are first uploaded and then replicated among the other repositories to increase the likelihood of finding the requested files in a timely fashion. The existing mobility of public transit buses is used to opportunistically synchronize files amongst the repositories.
- **Virual migration in vehicular cloud networks** (Section 2.2). The offloading spots are dematerialized as they now refer to areas where vehicles come in contact often and for long time periods. The benefit of identifying those areas is shown in the context of a virtualized large-scale mobile network consisting of the public buses operating in a large city. Buses are equipped with computation and storage resources virtualized so multiple tenants can deploy their services above the same mobile substrate. Vehicle contacts are used to migrate virtual machines hosted by the buses as a result of changes in the physical topology or in the allocation of their virtual resources.

Both work required the development of a framework including original methods for analyzing real mobility traces. This framework takes benefit of the increasing number of public transit agencies in large cities in North America or Europe releasing schedules and geographic information including stops, routes, trips, and other schedule data. Those information come as a collection of files following the General Transit Feed Specification (GTFS)[1] format. Our framework infers the real movements of public transportation vehicles such as buses we use to conduct experiment-based evaluations of the two aforementioned vehicular cloud services.

Based on those two extensions combined to our original work on traffic offloading, offloading spots can be seen as locations where the trajectories of multiple entities are combined in a travel path followed by the data. This observations led me to conduct a survey of the significant research taking advantage of various mobile entities to carry data either in replacement or in conjunction of a conventional data network.

- **Mobile-assisted data transmission systems** (Section 2.3). I present a classification of the existing strategies for composing the trajectories of multiple mobile entities. Trajectory composition may require the entities to meet at the same time so data can be passed synchronously or may involve an infrastructure where data is buffered by a first entity until passed later to a subsequent entity. Composition may also occur when two entities meet wherever they do or only if they meet in specific pre-defined locations. I describe the different approaches and methods defined so to realize each of those strategies.

2.1 | *Vehicular file storage and sharing system*

> (Publication [44]: B. Baron, P. Spathis, M. Dias de Amorim, and M. Ammar. "Cloud Storage for Mobile Users Using Pre-Positioned Storage Facilities." *ACM MobiCom Workshop on Experiences with Design and Implementation of Smart Objects (SMARTOBJECTS)*, 2016.)
> (Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)

In this section, I present the design of a cloud-like file storage and sharing system specifically targeting mobile users in urban scenarios. The system deploys a collection of strategically located pre-positioned data storage facilities acting as file repositories. Mobile users upload files that they would like to archive or share as those files can be later retrieved by the original uploaders or by other users from this set of repositories as they meet them. The existing mobility of the initial uploaders and subsequent users is exploited to limit the dependency of this system on conventional infrastructure-based networks.

The replication of files uploaded in the system results of the mobility of either the initial file uploader or subsequent users to replicate files. The distinctive feature of this approach lies in the opportunistic use of mobile users as data shuttles between the file repositories. Copies of the files are transferred to users when in the vicinity of a repository and

[1] https://developers.google.com/transit/gtfs/

transported to other repositories along user routes. To increase the likelihood of replication, copies of the user files can be loaded on multiple users traveling among the repositories.

One of the main challenges in implementing this system is how to handle user requests to either store a file or retrieve one in a timely fashion. A request is delayed depending on how close the user is from the nearest repository. Each request comes with a deadline indicating when it expires. Past the deadline, the request is considered as failed. The objective of the study presented below is to bound the request deadline to minimize the failure rate of requests.

The contribution of this work came into the form of a placement algorithm that determines the locations of a target number of repositories for a given user request deadline. The goals of the placement algorithm are twofold: (*i*) determine locations such that the allocated repositories serve the maximum number of user requests before they expire and (*ii*) connect the repositories together by the movements of the mobile users to create the network used for file distribution.

2.1.1 Repository placement model and resolution

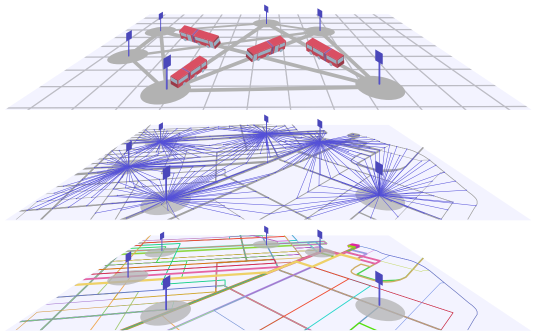


Figure C.1. Different layers to represent the requirements of the placement algorithm in the case of a bus transit system.

The requirements of the placement algorithm are represented in Figure C.1 in the context of a bus transit system. The bottom layer shows the trajectories of the buses in the Financial District of San Francisco and the subset of the bus stops acting as file repositories. Each bus line is represented by a color and a width indicating the frequency of buses operating on each line. The middle layer shows the discretized demands generated by the passengers of buses on their trajectories. This layer shows how the demands are allocated to the repositories depending on their respective distance. Finally, the top layer shows a logical graph where the nodes correspond to the repositories and the edges represented the logical path travelled by flows of buses running between two repositories with the purpose of synchronization. Similarly to the bottom layer, the width of the edges corresponds to the frequencies of buses.

The formulation of the repository placement is derived from a set cover problem, in particular, the Maximal Covering Location Problem [53]. Given a set of candidate locations for the repositories, the placement problem consists in selecting the candidate locations that maximize the success ratio of the requests issued by the users. This problem was shown to be NP-Hard [61], so we adapt known heuristics to solve it. To this end, we consider the Greedy Adding with Substitution (GAS) heuristic [53] that determines the optimal locations for each iteration of the algorithm.

2.1.2 Real-world data simulations

To evaluate our repository placement strategy, we first devise a realistic deployment of a predefined number of repositories by considering the stops and trips made by the buses operated by Muni[2], the public transit system of San Francisco. We first recreate the real movements of Muni buses and then, select the bus stops where to deploy the repositories.

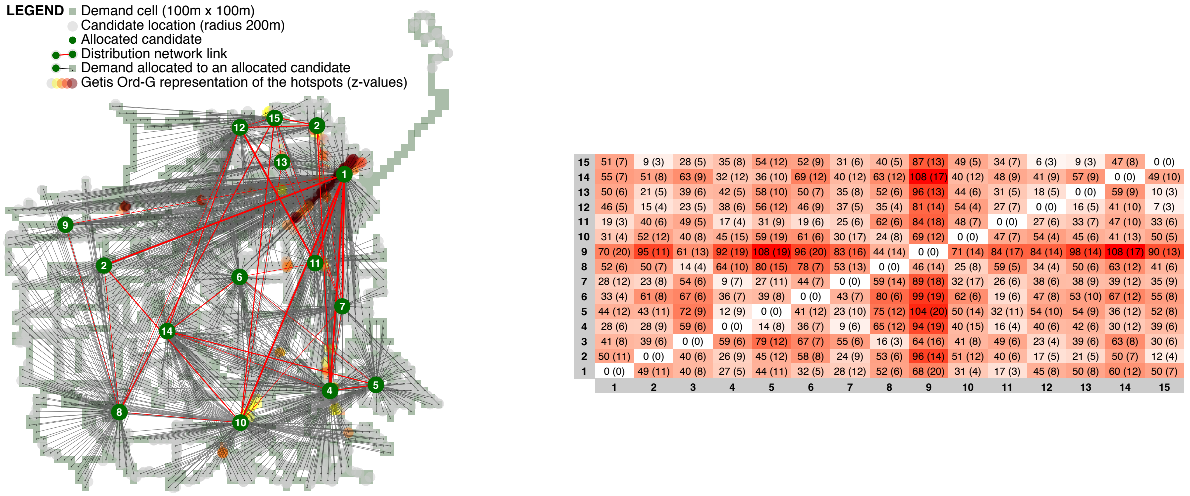
We do the following assumptions. The mobile users are passengers riding the buses. The repositories are equipped with IEEE 802.11 network interfaces which supports a radio range of 250 m. We ignore wireless interference and assume the capacity enough to accommodate the full exchange of files between the repositories and the bus passengers. The buses stop at the bus stops for a duration taken randomly between 10 seconds and 30 seconds.

No assumptions is made regarding the popularity and geographical locality of the files: A request is made every 10 minutes following a Poisson distribution. We distribute the requests uniformly across the buses, the trips they make, and the routes offered by the public bus transit service. We consider two types of requests according to the following ratios: 10% of the requests are requests to store a user file and 90% of the requests are to retrieve an available file. The simulation duration is 20,000 seconds.

Recreating San Francisco MUNI buses' movements. We developed a framework to analyze the mobility data from real-world traces. The dataset describing the bus movements consists of a set of files following the GTFS (*General Transit Feed Specification*) format[3]. In GTFS files, the mobility of a node is given by a collection of tuples consisting of the following fields: *<node id, timestamp, latitude, longitude>*. We developed tools to infer the complete bus movements from the GTFS files. The format of the output resulting of our tools is compatible with the ONE simulator [63]. We could recreate the movements of 493 buses resulting of 130 different trips made between 10am and 4pm on a typical weekday. We then use the ONE simulator to simulate the movements of the MUNI buses connecting the repositories hosted by the bus stops selected as described below.

[2] https://www.sfmta.com/

[3] https://developers.google.com/transit/gtfs


LEGEND
Demand cell (100m x 100m)
Candidate location (radius 200m)
Allocated candidate
Distribution network link
Demand allocated to an allocated candidate
Getis Ord-G representation of the hotspots (z-values)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	51 (7)	9 (3)	28 (5)	35 (8)	54 (12)	52 (9)	31 (6)	40 (5)	87 (13)	49 (5)	34 (7)	6 (3)	9 (3)	47 (8)	0 (0)
14	55 (7)	51 (8)	63 (9)	32 (12)	36 (10)	69 (12)	40 (12)	63 (12)	108 (17)	40 (12)	48 (9)	41 (9)	57 (9)	0 (0)	49 (10)
13	50 (6)	21 (5)	39 (6)	42 (5)	58 (10)	50 (7)	35 (8)	52 (6)	96 (13)	44 (6)	31 (5)	18 (5)	0 (0)	59 (9)	10 (3)
12	46 (5)	15 (4)	23 (5)	38 (6)	56 (12)	46 (9)	37 (5)	35 (4)	81 (14)	54 (4)	27 (7)	0 (0)	16 (5)	41 (10)	7 (3)
11	19 (3)	40 (6)	49 (5)	17 (4)	31 (9)	19 (6)	25 (6)	62 (6)	84 (18)	48 (7)	0 (0)	27 (6)	33 (7)	47 (10)	33 (6)
10	31 (4)	52 (12)	40 (8)	45 (15)	59 (19)	61 (6)	30 (17)	24 (8)	69 (12)	0 (0)	47 (7)	54 (4)	45 (6)	41 (13)	50 (5)
9	70 (20)	95 (11)	61 (13)	92 (19)	108 (19)	96 (20)	83 (16)	44 (14)	0 (0)	71 (14)	84 (17)	84 (14)	98 (14)	108 (17)	90 (13)
8	52 (6)	50 (7)	14 (4)	64 (10)	80 (15)	78 (7)	53 (13)	0 (0)	46 (14)	25 (8)	59 (5)	34 (4)	50 (6)	63 (12)	41 (6)
7	28 (12)	23 (8)	54 (6)	9 (7)	27 (11)	44 (7)	0 (0)	59 (14)	89 (18)	32 (17)	26 (6)	38 (6)	38 (9)	39 (12)	35 (9)
6	33 (4)	61 (8)	67 (6)	36 (7)	39 (8)	0 (0)	43 (7)	80 (6)	99 (19)	62 (6)	19 (6)	47 (8)	53 (10)	67 (12)	55 (8)
5	44 (12)	43 (11)	72 (9)	12 (9)	0 (0)	41 (12)	23 (10)	75 (12)	104 (20)	50 (14)	32 (11)	54 (10)	54 (9)	36 (12)	52 (8)
4	28 (6)	28 (9)	59 (6)	0 (0)	14 (8)	36 (7)	9 (6)	65 (12)	94 (19)	40 (15)	16 (4)	40 (6)	42 (6)	30 (12)	39 (6)
3	41 (8)	39 (6)	0 (0)	59 (6)	79 (12)	67 (7)	55 (6)	16 (3)	64 (16)	41 (8)	49 (6)	23 (4)	39 (6)	63 (8)	30 (6)
2	50 (11)	0 (0)	40 (6)	26 (9)	45 (12)	58 (8)	24 (9)	53 (6)	96 (14)	51 (12)	40 (6)	17 (5)	21 (5)	50 (7)	12 (4)
1	0 (0)	49 (11)	40 (8)	27 (5)	44 (11)	32 (5)	28 (12)	52 (6)	68 (20)	31 (4)	17 (3)	45 (8)	50 (8)	60 (12)	50 (7)

(a) Allocation of the repositories. The color variations of the demand and candidate locations represent hotspots (spatially significant locations) computed using the p-values of the average visit frequencies resulting from Getis Ord-G spatial analysis [62].

(b) Origin-destination matrix of the repositories (identified by the numbers represented on the Figure C.2a) showing the mean travel time in minutes between each pair of repositories. The values in parentheses are the corresponding standard deviation. The simulations were done for 15 repositories placed as shown in Figure C.2a.

Figure C.2. Results for the allocation of 15 repositories in San Francisco using the bus movements of the MUNI bus transit system.

Placing repositories in the city of San Francisco. The repositories are placed by an iterative procedure where each step finds the location of a bus stop for one repository. The first repository is placed at the bus stop capturing the most surrounding requests. A bus stop can satisfy a passenger request if the passenger reaches the radio range of the bus stop before the request expires. The satisfied requests and the bus stop selected in the current step is removed before moving one step forward. Each following step starts by compiling a list of candidate stops consisting of the top 25 % bus stops with respect to the number of requests they can satisfy. A bus stop is then selected out of those candidates depending on the capacity resulting from the flows of buses making trips connecting this stop to all the stops selected in the previous steps.

To simulate our placement procedure, we measure (*i*) the density of contacts between buses and the bus stops they visit and (*ii*) the frequency of the requests occurring in the same areas. We make a visual representation of those two variables using a heat map. We first divide the geographical area of interest in a grid pattern where each cell has a fixed size, e.g. 100m × 100m. We plot the aggregated values for each cell using different shades of color, the darker indicating higher densities. Heatmap representations help preprocess geo-dependent data which allows a finer time resolution of the placement procedure.

The output of our placement procedure for 15 repositories is shown on the heatmap of Muni bus contact depicted in Figure C.2a. The location of those 15 repositories was selected out of the 4,590 bus stops in San Francisco. The requests not yet satisfied 2,400 sec after being issued are considered as expired.

Distribution duration. One of the performance metric we evaluate is the time needed to distribute a file within all available repositories.

Figure C.3 shows the average duration to distribute a file within the network of repositories. We measure the distribution duration from the time a file is first uploaded in a repository and until the file is fully replicated among all other repositories. The repository where a copy of the file is first stored can be any of the repositories available.

We plot the distribution duration as a function of the file availability expressed as the percentage of repositories where a copy of the file can be found. This result is given for a total number of repositories varying from 5 to 25. For example, with 10 repositories, a file availability of .4 indicates that the file is available at four repositories.

We notice that the average duration to distribute the user files to all repositories, regardless of the number of repositories available, is 4000 seconds, or slightly more than one hour. It takes more time to distribute the copies of the files to the repositories at the beginning of the file distribution and at the end. At the beginning of the file distribution, only one copy of the

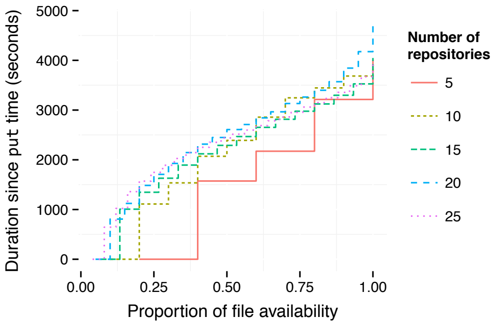


Figure C.3. Distribution duration as a function of the file availability for different numbers of repositories.

files is available at the first repository. It takes on average of 700 seconds to distribute the first copy of the file to the second repository. Then, as more repositories distribute copies of the files, the distribution of the copies is faster. At the end of the file distribution, copies of files are available at most repositories. It takes on average 500 seconds to reach the last repository, as it is the farthest away from the first repository where the original copy of the file was stored.

In Figure C.2b, we give an origin-destination matrix that shows the average travel time in minutes between any pair of repositories. This translates to the duration it takes to distribute the file to the different repositories after the first copy of a file was stored in a repository. These values give the average duration for a file to be available at a target repository, depending on the originated repository. In the figure, the repositories are identified by the same numbers as used in Figure C.2a. The connectivity between two repositories depends on the number and frequency of the buses whose trips connect one repository to another. For instance, repositories 1 and 15 are very well connected to the rest of the repositories. However, this is not the case of repository 9 since it is farther away from the rest of the repositories. This also further explains the longer duration it takes to distribute copies of the file from repository 9 to every other repository. Conversely, some repositories are farther away, which takes more time on average to reach them and distribute copies of the files to them.

Takeaway C.6

The scheduled mobility of surrounding entities such as public buses can be used to design a file sharing system with limited reliance on conventional data networks. Mobile users upload their files to the nearest bus stop acting as repository from where, each file is replicated among all other repositories using the buses' regular scheduled routes. A placement algorithm selects the bus stops where to install a repository so to maximize the ratio of successful upload and download requests while minimizing the time needed to satisfy these requests. The simulation results using real world mobility traces for San Francisco Muni public buses show that only 15 bus stops can cover the requests wherever issued in almost one hour.

2.2 | *Virtual machine migration in vehicular cloud networks*

(Publication [45]: B. Baron, M. Campista, P. Spathis, L. H. MK Costa, M. Dias de Amorim, O. Carlos MB Duarte, G. Pujolle, and Y. Viniotis. "Virtualizing vehicular node resources: Feasibility study of virtual machine migration." *Elsevier Vehicular Communications*, 2016.)
(Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)

In this section, I address the problem of virtualizing the resources of large-scale mobile networks. This work focuses on the migration of virtual machines triggered by the reallocations of virtual resources or changes in the physical topology. Instead of using cellular connectivity, the originality of this work lies in the use of V2V communications between vehicles to migrate the virtual machines. In this work, I study the contact opportunities resulting of the trips performed by the public buses operating in large cities.

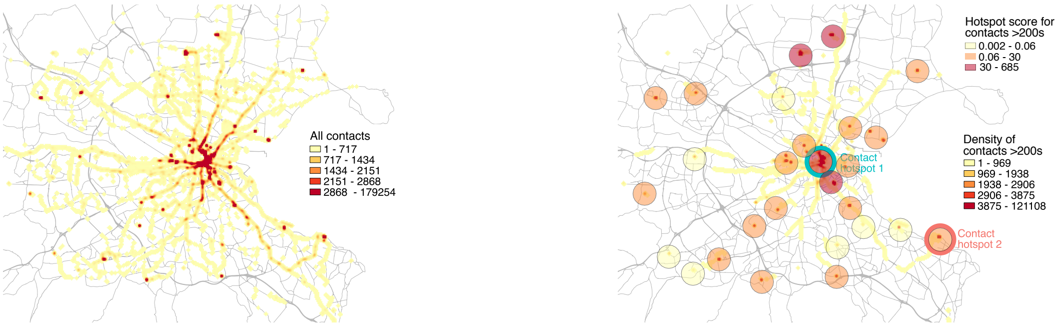
This work draws on the results presented for various cities including San Francisco and Warsaw where it has been shown that vehicles come in contact more often and for longer period of time in specific areas [64]. By limiting the virtual machine migrations to those specific areas, the expected benefit is an improved ratio of successful migrations. We analyze the contact opportunities between the public transit buses of the city of Dublin by following a methodology similar to the one we used in Section 2.1. This methodology helps evaluate the opportunities of migrating VMs between buses in contact while moving.

2.2.1 Contact density heatmap generation.

The methodology consists first in processing the mobility traces with the purpose of generating a heatmap representing graphically the spatial variation of the contact density. The next steps consist in determining the location of the hotspots matching specific values regarding the duration and the rate of the contacts occurring in those areas.

Contact heatmap. The first step consists in computing the spatial variations of the contact density. The geographical area of interest is divided in a grid pattern where square cell has a fixed size, *e.g.*, 100m × 100m. In each cell, we start by plotting all the contacts whatever their durations as presented in Figure C.4a. We then plot Figure C.4b by selecting the only contacts relevant with respect to our virtual machine migration scenario. Those are the contacts that last for at least 200 sec. This duration amounts for the time needed to transfer a typical virtual machine of 200 MB. A kernel density estimator is then used to characterize the contact density of each cell. This consists in clustering the points representing the contacts occurring in the cell. Larger numbers of clustered points result in larger contact density values.

The heatmap obtained for the city of Dublin is shown in Figure C.4b where different contact densities are depicted using different shades of red, the darker shade indicating a higher density of contacts. We notice that the contacts between buses are concentrated in the city center, but also on the outskirts of the city center, along the major traffic arteries and at some of the main intersections of Dublin. The next steps consist in deriving the hotspot locations from the contact density heatmap.



(a) Heatmap of all the contacts whatever their duration between buses operating in Dublin, Ireland.

(b) Heatmap of the contacts whose duration is greater than 200 seconds. The hotspots highlighted by circles contain the aggregated cells where 25% of all contacts occur.

Figure C.4. Heatmaps of contacts between buses in Dublin, Ireland. The figures denote the number of contacts per cell (of size 100m × 100m). We choose to simulate migrations between buses entering the two hotspots circled on the map of Figure C.4b.

Contact hotspots. Firstly, the cells are ranked according to their contact density. To exclude cells where contacts are not occurring in significant numbers, the 25% top cells are selected. The contiguous cells are then aggregated in clusters referred to as *hotspots*. Hotspots are represented on the map by circles centered so they can cover the corresponding cluster. The diameter of the hotspots is fixed and selected so the largest cluster is fully covered by a single hotspot.

The hotspots depicted in Figure C.4b are further characterized from the perspective of the V2V migration of virtual machines. For each hotspot, the two following variables are computed: The total number of contacts with a duration of at least 200 sec (n) and the average inter-contact duration (x). The inter-contact duration refers to the duration between two consecutive contacts occurring within the same hotspot. Those two variables are then used to assign a score to each hotspot we compute as: ${}^{n}\!/_{\max(1,\,x)}$. Finally, we use the Jenks optimization method to identify three value ranges which provide the best arrangement of the hotspots according to their scores.

In Figure C.4b, different shades of color are used to represent the hotspots according to their classes. In the performance evaluation, two significant hotspots are selected, each belonging to the two higher classes: "Hotspot 1" has the highest score which indicates that long-duration contacts occur with high frequency. Note that three other hotspots share a similar score. The second hotspot indicated as "Hotspot 2" is selected arbitrarily among the 16 occurrences sharing a similar score.

2.2.2 Real-world data experimentation

This section presents the results of simulations conducted to assess the feasibility of migrating virtual machines (VM) in the context of the public transit system of Dublin city. The focus of those results are on the capacity of the opportunistic contacts between buses while in transit.

Experimental setup. For our feasibility study, we use a publicly available dataset of real traces of Dublin city buses provided by Dublinked, part of the Dublin City Council. The mobility traces span the month of January 2013 and provide timestamped GPS coordinates of all DublinBus buses in service [65]. We analyze the movements of 823 buses during the service day of Tuesday, January 29, 2013 from 10am to 1pm. We assume that two buses are in contact whenever the buses are in each other's communication radius denoted R.

To reproduce the movements of DublinBus buses and infer their contacts, we use the ONE simulator [63]. We used the default settings of the ONE to simulate IEEE 802.11 on the mobile nodes. To account for the properties of the physical layer including the link-level packet losses, we considered a conservative communication radius R of 150 meters with an homogeneous average goodput of 1 MB per second.

VM migration simulations. We simulate the migration of virtual machines of various sizes running on buses entering the two hotspots circled on the contact heatmap of Dublin (see Figure C.4b). We make no prediction regarding the duration of the contacts: A bus entering one of those hotspots initiates the migration of a virtual machine with the first bus it comes in contact with. The migration is aborted if the contact duration is not long enough to accommodate the transfer.

The results we present reflect averages over 100 virtual machine migration trials for different sizes of virtual machines we try to migrate at each of the two hotspots. The range of values we use captures the usual sizes of virtual machines that can be found, that is, from a few hundreds of kilobytes (*e.g.*, TinyOS [66]) to a few hundreds of Megabytes [67]. For each size of virtual machine, we measure the average time needed for a bus entering each hotspot to find a suitable contact and the time it actually takes to migrate the virtual machine. The plot in Figure C.5 shows the mean time to transfer virtual machines of different sizes, as well as the proportion of virtual machines that were actually transferred.

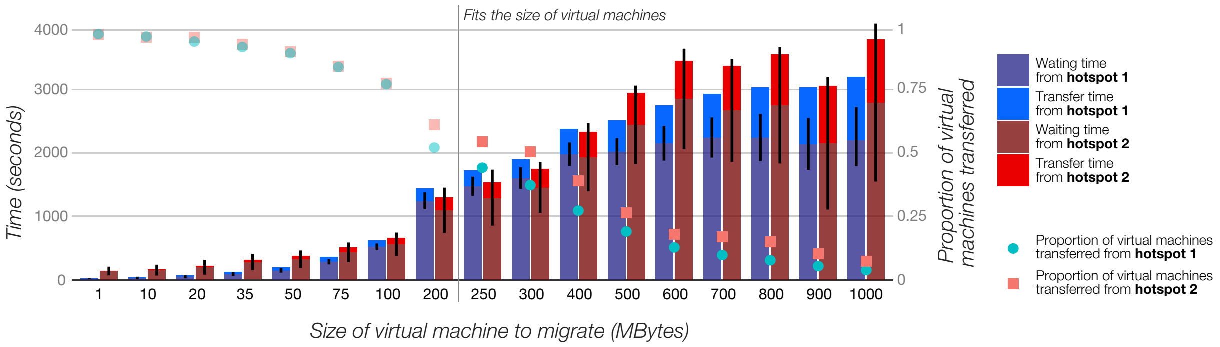


Figure C.5. Proportion of successful virtual machine migrations and mean time to transfer the virtual machine as a function of the virtual machine size. The black solid bars represent the 95% confidence interval for the waiting time before successfully transferring the virtual machines.

First, we note that the ratio of virtual machines successfully migrated at both hotspots decreases as their size increases. This is due to the lack of enough long-lasting contacts between buses. Buses fail several times before finding a suitable contact to transfer the virtual machine. This observation is confirmed by the time spent waiting for a contact suitable in duration with large-sized virtual machines. Overall, it takes less time to find a suitable contact at Hotspot 1 than at Hotspot 2. Buses entering Hotspot 1 spend less time waiting for a suitable contact in comparison with Hotspot 2 expect for virtual machine sizes ranging from 200 to 400 MB.

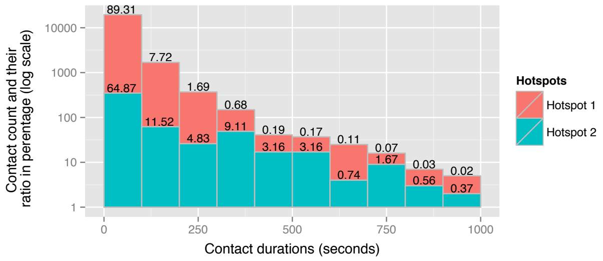


Figure C.6. Number of contacts in log scale accounted for different duration ranges and their ratios (in percentage) against the total number of contacts in the Hotspots 1 and 2, respectively.

To further explain this trend, we plot in Figure C.6 the total number of contacts measured for different contact durations as well as the ratios of those contacts against the total number of contacts occurring in Hotspots 1 and 2. We can see that Hotspot 2 has a higher ratio of contacts lasting long enough (e.g. 200 sec and 400 sec) to accommodate transfers of virtual machines with a size ranging from 200 MB to 400 MB. Despite the higher number of contacts at Hotspot 1, most of them are not suitable for the transfers of such virtual machines. Buses spent more time trying transfers that will eventually fail before finding a suitable contact. This also explains why the ratio of successful migrations is higher at Hotspot 2 when virtual machines have sizes ranging from 200 to 400 MB.

Takeaway C.7

The virtualization of a large scale mobile network consisting of the public buses operated in large cities can be used to deploy a wide range of concurrent services for urban sensing or intelligent transportation. To address the need of reallocating the network virtual resources, virtual machines are migrated using the opportunistic contacts between buses in replacement of traditional cellular wireless networks. Our simulation results using real world mobility traces for Dublin public buses show that virtual machines of several hundreds of Megabytes can migrate between moving buses.

2.3 | *Classification of the strategies for composing mobile entities trajectories*

(Submission: B. Baron, P. Spathis, M. Dias de Amorim, and Y. Viniotis. "Mobility as an alternative communication channel: A survey." Submitted to *IEEE CommunicationsSurveys & Tutorials*, 2016.)
(Ph.D. thesis of Benjamin Baron, UPMC Sorbonne Universités, 2016 [43]. Co-supervised by Marcelo Dias de Amorim.)

The emergence of wireless capabilities equipping an ever-growing range of mobile entities has led to various ad hoc paradigms such as Mobile Ad hoc Networks (MANET), Vehicular Ad hoc Networks (VANET), Wireless Sensor Networks (WSN), and Delay-Tolerant Networks (DTN). One of the differences in those paradigms lies in how data is forwarded toward the final destination. In MANET, for example, forwarding is used in its commonly understood meaning and performed in a similar way as in wired networks: data is forwarded along a path resulting from contemporaneous end-to-end connectivity. Other paradigms somehow acknowledge the fact that node encounters intersperse mobility phases which also contribute to the movement (i.e., forwarding) of the data. This is the case with DTNs and their close

Table C.1: A classification of data delivery strategies, along with their research directions and surveyed works.

Direct data delivery	**Indirect data delivery**			
		Asynchronous data passing		*Synchronous data passing*
Passive method Random: [70], [71] Predictable: [80], [81], [82] Scheduled: [91], [92], [93] *Active method* Single: [103], [104], [105] Multiple: [102], [109], [110]	*Stationary*	Non-controlled mobility [83], [84], [85], [86], [87], [88], [89], [90] Controlled and paid mobility [100], [101], [102]	*Floating*	No control plane [71], [72], [73], [74], [75], [76], [77], [78], [79] Local control plane [94], [95], [96] [97], [98], [99] In-band global control plane [106], [107], [108], [99]
Paying method Postal services: [100], [115] Commercial: Netflix, AWS	*Mobile*	Centralized global control plane [111] (NIMF and FIMF), [112] Distributed global control plane [116], [117]	*Pre-defined*	In-band global control plane [113], [64], [114] Out-of-band global control plane [102], [118], [119]

variants, such as Opportunistic Mobile Networks (OMN), which introduce the so-called store-carry-and-forward principle [68]. The rationale behind this principle is to exploit node mobility with a view to increasing contact opportunities in low-density networks.

Based on this observation, we proposed to survey the relevant literature which focuses on the benefits of the *carry* phase enabled by the mobility of a wide variety of entities. The scope of this survey include the alternative data transmission methods resulting from piggybacking storage devices on mobile entities. Informally referred to as Sneakernets [69], those methods exploit the movements of mobile entities to transfer data in replacement to or to supplement conventional transmissions over computer networks. Typical use-case scenarios and applications are concerned with specific types of data that can tolerate delay in their delivery. Those scenarios involve a wide range of entities that are mobile either by nature, such as humans or animals, or by conception, with engine-powered vehicles or robots.

Depending on the entities involved and the services exploiting their movements, the expected benefits of these alternative transmission methods depend on various parameters. Examples of such parameters are the total number of mobile entities and the size of the memory they carry, the level of knowledge regarding their mobility, or at what extent their mobility can be controlled. A common goal of the works we review is to alleviate the limited control (or the lack of knowledge) concerning the mobility of the entities, especially when these entities move for other purposes than the intended data exchange.

Direct data delivery. We first review the approaches that exploit the mobility of one or more independent entities to carry data between a source and a destination. The main difference between those approaches lies in the degree of randomness in the movements of the entities they employ. We distinguish the approaches that passively exploit the existing mobility of entities moving for other purposes than delivering the data they carry from those that rely on controllable entities. In the latter case, the entities follow trajectories calculated for the specific purpose of data delivery. The entities can also refer to the fleet of vehicles operated by a postal or package delivery company in charge of transporting the data.

Indirect data delivery. We then review the approaches that involve a sequence of mobile entities who take turns in delivering the data. The data takes a logical path, also called a vector route, consisting of multiple segments of trajectories, each followed by different mobile entities. Data is passed from one entity to another as a result of a process referred to *forwarding* when entities are in direct contact. We propose a classification of those approaches based on the strategy they use for composing the trajectories of the different entities. Those strategies differ according to the two following criteria:

1. The *time* when composition occurs. The data can be passed either synchronously at the time two entities meet or buffered at specific locations before passed asynchronously to subsequent entities.
2. The *location* where composition is performed. The location can be either pre-defined or floating. In the pre-defined case, the data is passed only if the entities are in contact at specific locations. In the floating case, composition can result of contacts between entities wherever they meet.

The classification we proposed, results of the combinations of these two criteria instantiated according to the strategies we review. Each resulting class can be found in Table C.1. For each class, I provide the bibliographic citations to the various work that were reviewed.

Sub-level classification. We further classify the two previous delivery models as follows:

- **Direct data delivery.** We classify the approaches based on the direct delivery model according to the three following methods:
  - The *passive method* consists in using opportunistically the existing mobility of a single or a group of independent entities called data mules to transport data. The performance of this method depends on the degree of knowledge of the mobility of the entity which may result from the predictability of their movements.
  - The *active method* consists in controlling the mobility of entities commonly referred to as data ferries. These entities follow precalculated routes with the purpose of improving the data delivery rate and delivery latency.
  - The *paying method* consists in using the paid services of a postal or delivery company. This method is a combination of the two previous methods: The delivery trucks may be considered as controllable entities, nevertheless the calculation of their routes is integrated to the service provided by the delivery company. Furthermore, the use of this mobility results of an active method since delivery services are purchased with the obvious purpose of transporting data.
- **Indirect data delivery.** We classify the approaches following the indirect model depending on the method they introduce to pass the data between two consecutive entities:
  - **Asynchronous passing:** This method consists in passing the data indirectly via stationary or controllable mobile nodes. The stationary nodes act as data exchange relay points where entities can drop off data for later pick-ups by another entity. The movements of controllable nodes allow the data to be passed between two separated entities following non-intersecting routes.
    - *Stationary intermediate nodes.* A stationary intermediate node allows the data to be passed between two entities whose trajectories intersect without the entities being in contact at the same time. The intermediate nodes buffer the data so it can be passed asynchronously from one mobile entity to another.
    - *Mobile intermediate nodes.* In the case of entities following trajectories that intersect occasionally or never, the use of mobile intermediate nodes, such as ferries or robots can bridge the gap between such entities. The route of a mobile intermediate node is calculated so the data can be passed asynchronously at various locations depending on the meeting points between the nodes and the entities.
  - **Synchronous passing:** This method consists in passing the data directly from one entity to another while physically in contact. The objective is to decide whether the data should be passed or not every time two entities are in direct contact. As a result, the data can be passed to the first entity encountered, to a subsequent one, or to more than one encountered entities. The latter case results in data replications intended to improve the delivery likelihood. We classify the approaches that use synchronous indirect delivery depending on whether the data can be passed anywhere as long the entities meet or when they meet at pre-defined locations:
    - *Floating composition.* In the floating case, the data can be passed anywhere as long the mobile entities are in direct contact. The decision whether to pass the data or not results of strategies, if any, which consists in limiting the number of copies resulting from passing the same data at the occasion of multiple contacts.
    - *Pre-positioned composition.* In the pre-positioned case, the decision to pass the data is taken when mobile entities meet at specific locations. These locations are determined given some specific properties such as the contact density.

2.4 | Conclusion

This chapter has investigated the benefits of exploiting the mobility of public transit buses in urban scenarios. We proposed two cloud related services with limited dependency on conventional data networks. The first service is a file storage and sharing system consisting of a collection of bus stops equipped with wireless data storage devices. Bus stops so equipped act as file repositories where passengers can upload or retrieve a file via short-range radio.

In a second service, we removed the need of an infrastructure as buses can now exchange data while in contact. The lack of infrastructure support implies synchronous exchange of data in a similar way as with the DTN paradigm. Whereas DTNs allow data to be passed wherever nodes are in contact, we limit data exchanges to pre-positioned areas where the likelihood that buses meet often and long enough is higher compared to other areas.

For both services, we developed a framework for analyzing real-world mobility trace was developed. This framework includes new methods to characterize the contact density of geographical areas according to the movements of buses crossing those areas. We identified mobility patterns resulting of both the consistency between bus trips and the structure of the routes they follow.

The various methods used in these two cloud services to compose the trajectories of multiple mobile entities provided us with the premises for a survey of the research literature taking advantage of mobile entities carrying data while they move. We identified the various methods used to pass data between mobile entities whose movements are combined in various scenarios, including bridging connectivity gaps or deploying ad hoc networks in challenging environments. We have reviewed the significant approaches in the way they address the challenges resulting from the partial knowledge regarding entities' mobility or connectivity.

Part D

Summary and Outlook

In this chapter, I first present the general conclusions of this document and then list a number of perspectives for future work.

Summary

In this thesis, I have presented a selection of the research work I have been conducting since 2008 as an assitsant professor at UPMC. Some of this work was conducted in productive research areas I identified or developed for their potential for research contributions. My work lead to research outcomes including publications and also the development of research projects that I have participated to or lead, at both national and international level. Those projects were the result of writing the proposals and their successful application to research funding. The research work reported in this thesis is the result of activities which amount, as established by ministerial decree, for half of my workload as a faculty at a French university. The rest of my working hours is devoted to teaching activities but also administrative tasks and community services which, for some, are related to research. Regarding teaching, I always took the opportunity of teaching for undergraduate and graduate students to explain my research interests and topics as a way of raising their awareness of research.

Along the course of my carrier, my contributions came into the framework of various research areas. These areas have been defined for and with the Ph.D. candidates I co-advised so as to support their advancement in research and secure their achievements by mean of publications. Each area has been chosen to minimize conflicts between the expected outcomes of the funding research projects and the research goals of the Ph.D. candidates as well as their background and skills. The research topics presented in this thesis can be directly related to the design and the analysis of information delivery systems. The main driving motivation of my work was to tackle mobility by considering the many forms it can take. I consider in turn, the mobility of content when migrated or replicated between fixed repositories, the mobility of the users as the content recipients, and the mobility of vehicles as the content carriers. I addressed those various scenarios by considering mobility either as a challenge or as an enabler.

In a first part of this thesis, I have presented the results of a work I conducted by following a conventional approach to research: I address the limitations of existing network architectures and the challenges emerging from new architectural alternatives. First, I addressed the mismatch between the host centric design of the Internet and the needs of content related services. I proposed to extend the CCN routing framework with a query-based routing protocol that leverages the replication of popular content. Each content item and its replicas are provided with a location-independent identifier which allows handling content replication in a natural way. The novelty of my approach lies in the use of the caching capabilities of content routers to store the routes discovered toward the closest content replicas. Next, I took account of the user need in accessing data while on the move. Instead of perceiving the mobility as a challenge, I exploit user movements as a radio resource saver in cellular networks. I proposed a content dissemination strategy that leverages node mobility and delay-tolerance to determine the best possible times an access point should transmit a copy of a given content. In a nutshell, its goal is to let users gather within fewer cells to reduce the number of transmissions. I then address the problem of data dissemination in vehicular networks by first proposing a component-based analysis of the most relevant protocols proposed in the literature. I identified a set of generic building blocks and compared the performance of their respective technical realizations giving various applications profiles and environment parameters. Based on the results drawn from this analysis, I proposed the design of Servus, a broadcast-based dissemination protocol that efficiently tackles the challenges resulting from the dynamic structure of vehicular networks.

In the second part of my thesis, I have presented the work I started in 2009, when my research took the U-turn which got me landed in the domain of mobility-assisted, infrastructure-less data communications. This direction was motivated by the need to develop a longer term vision regarding the role of the connected car in various disruptive scenarios. Following up my work on vehicular networks, I decided to tackle mobility from a broader perspective. In my vision, mobile entities such as vehicles and the roads they travel form an ad hoc communication infrastructure that can be used in replacement of or to complement conventional data networks. To accomplish this vision, I decided to characterize mobility in terms of networking quantities and to find means of mitigating the randomness of vehicle mobility. My first contribution to this work came in the domain of traffic offloading. My objective was to exploit the routine daily journeys made by car which represent a tremendous amount of resource that can be used to move data *en masse*. I proposed a centrally-controlled architecture for offloading bulk data transfers from the Internet on the roads traveled by vehicles equipped with data storage devices. Data is stowed away on the vehicles so equipped, while been parked unbeknownst to their driver. To maximize the bandwidth resulting of the vehicles combined memory storage, a collection of wireless data storage equipments called offloading spots are deployed at locations where vehicles usually parked. An offloading spot acts as a data exchange relay point where a vehicle can drop off data for later pick-up by another vehicle depending on their direction of travel with respect to the destination of the data transfer. In a second piece of work, I exploit the storage capability of the offloading spots to design of vehicular-based cloud system for storing and sharing files. The offloading spots act as repositories where files are first uploaded and then replicated among the other repositories to increase the likelihood of finding the requested files in a timely fashion. The vehicles are here the public transit buses operating in a city and their mobility is used to opportunistically synchronize files amongst the repositories collocated in the bus stops. In a last work, the offloading spots are dematerialized as they now refer to areas where vehicles come in contact often and for long time periods. The benefit of identifying those areas is shown in the context of a virtualized large-scale mobile network consisting of the public buses operating in a large city. Vehicle contacts are used to migrate virtual machines hosted by the mobile nodes as a result of changes in the physical topology or reallocations of the virtual resources offered by the mobile nodes.

Perspectives

In this document, I studied different facets of mobility with regard to data delivery. The guiding principle to my work was to exploit the existing mobility of motorized entities such as private cars or public buses in the context of various mobility-assisted transmission scenarios. To overcome the different level of knowledge regarding the mobility of each entity, we developed various tools for some inspired from transportation research. Most of the presented results show that the dependence on conventional infrastructure-based networks can be mitigated and some of their limitations overcome. Mobility can be managed as a resource that can be used to offer services incorporating new semantics. This work opens many avenues for research that I describe for some in this last section. These avenues are in line with the new trends we observe in the automotive industry and the emerging economic opportunities toward car electrification, autonomous driving, and shared mobility.

Big data analytics for the car. In this thesis, I addressed the delivery of data not related to the vehicle or its driver and passengers. With the automation trend of driving and the need for information about the vehicles' environment and the vehicle itself, valuable data will be made available on the vehicle. Such data will unlock new revenue opportunities for car makers while improving previous processes such as car sales and marketing or dealer servicing. Given the market share of large car manufacturers, the ratio of cars on the road branded by a single manufacturer account for several hundred millions and the number of total miles travelled for one trillion miles per year. The data stream outputted by the hundreds of sensors that will soon equip, if not already, a vehicle is estimated in the 25 Gbs of data range per car per hour. Car manufacturers can monetize this stream of data by giving access to third-party companies or by offering in-house services. The ability to cope with data of such scale represents a major challenge with regard to vehicle connectivity and analytics. Car generated or captured data can be processed so to optimize the travel time and experience or to provide content and services through better driver or passenger profiling.

Mobility as a service. The significant amount of data that results from sensors, positioning systems, cameras, or other image acquisition systems combined to vehicle connectivity and intelligent analytics will provide with new opportunities such as real-time traffic or parking, refined navigation, or new mobility services. In addition to on-demand autonomous transportation, new ownership and usage models are been considered in a short term. According to recent studies, car manufacturers will shift from a one-time purchase model to the offering of new mobility services following a transaction-based model such as peer-to-peer carsharing or pay-as-you-go contracts. Such services can also provide users with more transportation options by combining multiple modes of transportation. The success of those services depend on how they combine multiple data sources such as traffic sensors or public transport data to best fit the preference of the users. In the work presented in this thesis, most of the datasets came in the form of road traffic count reports. We are currently studying means of correlating and processing real-time traffic data from diverse sources regarding many different modes of transportation.

Formation and management of ad hoc fleets. One interesting direction we are currently investigating is the ability to create ad hoc fleets of vehicles to offer services with new semantics. Third-party service providers or the car manufacturers themselves can take advantage of the advanced electronics such as sensors or real-time vehicle monitoring systems available on the vehicles to design new services without actually owning the cars. The driver or the car owner if different, may subscribe to a "get paid to drive" program and receive a monthly fee or a discount on the cost of charging their vehicle in exchange of driving their normal routine. The discount rate can be calculated based on the driving patterns including coverage and mileage. If the car manufacturers take on the role of service provider, vehicles are equipped as standard with required electronics and service is provided without involving or compensating the vehicles' drivers. If different, service providers can provide the car dongles while outsourcing hardware manufacturing to third-party companies. Possible services may follow the same semantic as real-time ridesharing but for the transportation of any movable items from one location to another. Advancements in real-time traffic monitoring can help existing transport companies already owning a fleet of vehicles in their need for business diversification by increasing the variety of services they offer, without being restricted to either goods or passenger transportation.

Decision-making oriented geo-nested analysis. To be able to take full advantage of the vehicle-generated or -captured data, vehicle computation should be involved in real-time analysis. Factors that may be considered include the vehicle characteristics, location, and travel log or the driving behavior and sociability of the drivers. Once processed, data may then be aggregated and collected in a DTN fashion without involving existing communication infrastructures. Data analysis may result of building blocks hosted by different vehicles connected together as vehicles come in contact. A similar scenario may be applied to the data collected by sensors on the vehicle or located by the roadside. Vehicles can process sensor data on-the-go once generated. Some vehicles may act as temporary sinks where data is collected from other vehicles. Vehicles may use opportunistic contacts so the data can gathered at the sinks or so the resulting decisions can be taken where needed. The decision to pass the data from one vehicles to another may result of dynamic floating forwarding policies computed by a controller for specific predefined locations. The controller enforces such policies through a control channel such as cellular connections and combine various attributes of the vehicles as they meet in those locations.

References

[1] Y. Barouni and P. Spathis, “Content centric routing for the autonomic networks,” in *2nd International Conference on Autonomic Computing and Communication Systems (Autonomics)*, 2008.

[2] Y. Barouni, P. Spathis, and S. Fdida, “Algorithme distribué de filtrage et de routage orienté contenu,” in *10ème Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*.

[3] Y. Barouni and P. Spathis, “Magnet: A content centric routing protocol for large scale networks,” in *4th International Conference on Future Internet Technologies (CFI)*, 2009.

[4] P. Spathis and M. Bicudo, “ANA: Autonomic Network Architecture,” in *Autonomic Network Management Principles: From Concepts to Applications*, pp. 49–65, Academic Press, 2010.

[5] M. Diallo, S. Fdida, and P. Spathis, “Towards extreme scale content-based networking for the next generation internet,” *IEICE transactions on communications*, vol. 94, no. 10, pp. 2706–2714, 2011.

[6] P. Spathis, S. E. Belouanas, and K.-L. Thai, “Leveraging replication in content-centric networks,” in *Proc. Global Information Infrastructure and Networking Symposium GIIS*, (Guadalajara, Mexico), Oct. 2015.

[7] S. E. BELOUANAS, T. Kim-Loan, P. Spathis, P. Spathis, M. Dias De Amorim, F. Rousseau, and A. Duda, “Content Centricity in Constrained Cellular-Assisted D2D Communications,” in *Ad Hoc Networks*, 2014.

[8] R.-A. Gorcitz, P. Spathis, M. Dias de Amorim, R. Wakikawa, and S. Fdida, “Dissecting dissemination in vanets,” in *ACM Asian Internet Engineeering Conference (AINTEC)*, 2012.

[9] R. A. Gorcitz, P. Spathis, M. D. de Amorim, R. Wakikawa, and S. Fdida, “SERVUS: Reliable low-cost and disconnection-aware broadcasting in VANETs,” in *IWCMC*, 2011.

[10] R. A. Gorcitz, *From Small-scale Data Dissemination to Large Data Transfers in Vehicular Environments*. PhD thesis, UPMC Sorbonne Université s, 2013.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *Proc. 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT’09)*, (Rome, Italy), pp. 1–12, 2009.

[12] M. Walfish, H. Balakrishnan, and S. Shenker, “Untangling the web from dns,” in *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI’04)*, (San Francisco, CA, USA), Mar. 2004.

[13] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, “Multiprotocol extensions for bgp-4,” Tech. Rep. 4760, Jan. 2007.

[14] R. Chiocchetti, D. Rossi, and G. Rossini, “ccnsim: An highly scalable ccn simulator.,” in *Proc. IEEE International Conference on Communications (ICC’13)*, pp. 2309–2314, IEEE, June 2013.

[15] V. Kumar, L. Lin, D. Krajzewicz, F. Hrizi, O. Martinez, J. Gozalvez, and R. Bauza, “itetris: Adaptation of its technologies for large scale integrated simulation,” in *Proceedings of the 71st Vehicular Technology Conference (VTC 2010-Spring)*, pp. 1–5, IEEE, 2010.

[16] F. Aurenhammer, “Voronoi diagrams—a survey of a fundamental geometric data structure,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 345–405, 1991.

[17] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of computer computations*, pp. 85–103, Springer, 1972.

[18] V. Chvatal, “A greedy heuristic for the set-covering problem,” *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[19] P. Slavík, “A tight analysis of the greedy algorithm for set cover,” in *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC)*, pp. 435–441, ACM, 1996.

[20] L. A. Wolsey, “An analysis of the greedy algorithm for the sub-modular set covering problem,” *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[21] T. Little and A. Agarwal, “An information propagation scheme for VANETs,” in *ITSC*, 2005.

[22] M.-T. Sun, W.-C. Feng, T.-H. Lai, K. Yamada, H. Okada, and K. Fujimura, “GPS-based message broadcast for adaptive inter-vehicle communications,” in *VTC*, 2000.

[23] B. Karp and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in *ACM Mobicom*, 2000.

[24] K. Suriyapaibonwattana and C. Pomavalai, “An effective safety alert broadcast algorithm for VANET,” in *ISCIT*, 2008.

[25] A. Benslimane, “Optimized dissemination of alarm messages in vehicular ad-hoc networks (VANET),” in *HCNMC*, 2004.

[26] M. Heissenbütell, T. Braun, T. Bernoulli, and M. Wälchli, “BLR: beacon-less routing algorithm for mobile ad-hoc networks,” vol. 27, pp. 1076–1086, July 2004.

[27] Q. Yu and H. G., “Abiding geocast for warning message dissemination in vehicular ad hoc networks,” (Beijing, China), May 2008.

[28] M.-T. Sun, W.-C. Feng, T.-H. Lai, K. Yamada, H. Okada, and K. Fujimura, “GPS-based message broadcast for adaptive inter-vehicle communications,” (Boston, USA), Sept. 2000.

[29] M. O. Cherif, S.-M. Secouci, and B. Ducourthial, “How to disseminate vehicular data efficiently in both highway and urban environments?,” (Niagara Falls, Canada), Oct. 2010.

[30] M. Slavik and I. Mahgoub, "Stochastic broadcast for VANET," (Las Vegas, NV, USA), Jan. 2010.

[31] M. Mariyasagayam, T. Osafune, and M. Lenardi, "Enhanced multi-hop vehicular broadcast (MHVB) for active safety applications," (Sophia Antipolis, France), June 2007.

[32] G. Korkmaz, E. Ekici, F. Özgüner, and Ü. Özgüner, "Urban multi-hop broadcast protocol for inter-vehicle communication systems," (Philadelphia, USA), Sept. 2004.

[33] M. Durresi, A. Durresi, and L. Barolli, "Emergency broadcast protocol for inter-vehicle communications," (Fukuoka, Japan), July 2005.

[34] T. Little and A. Agarwal, "An information propagation scheme for VANETs," (Boston, MA, USA), Sept. 2005.

[35] A. Agarwal, D. Starobinski, and T. Little, "Exploiting downstream mobility to achieve fast upstream message propagation in vehicular ad hoc networks," (Anchorage, AK, USA), May 2007.

[36] A. Bachir and A. Benslimane, "A multicast protocol in ad hoc networks inter-vehicle geocast," (Jeju, Korea), Apr. 2003.

[37] D. Sormani, G. Turconi, P. Costa, D. Frey, M. Migliavacca, and L. Mottola, "Towards lightweight information dissemination in inter-vehicular networks," (New York, NY, USA), Sept. 2006.

[38] J. Tian, J. Hähner, C. Becker, I. Stepanov, and K. Rothermel, "Graph-based mobility model for ad-hoc network simulation," (San Diego, USA), Apr. 2002.

[39] B. Baron, P. Spathis, H. Rivano, and M. D. de Amorim, "Étude de l'intermodalité pour le délestage des réseaux d'infrastructure," in *ALGOTEL 2015—17èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, 2015.

[40] B. Baron, P. Spathis, H. Rivano, and M. D. de Amorim, "Offloading massive data onto passenger vehicles: Topology simplification and traffic assignment," *IEEE/ACM Transactions on Networking*, 2015.

[41] R. A. Gorcitz, Y. Jarma, P. Spathis, M. D. de Amorim, R. Wakikawa, J. Whitbeck, V. Conan, and S. Fdida, "Vehicular carriers for big data transfers (poster)," in *IEEE Vehicular Networking Conference*, Nov. 2012.

[42] S. Fdida, P. Spathis, and M. D. de Amorim, "Massive data migration on wheels," in *WONS 2011—IFIP/IEEE International Conference on Wireless On-demand Network Systems and Services*, 2011.

[43] B. Baron, *Intermodal data transport for massive offloading of conventional networks*. PhD thesis, UPMC Sorbonne Université s, 2016.

[44] B. Baron, P. Spathis, M. Dias De Amorim, and M. Ammar, "Cloud Storage for Mobile Users Using Pre-Positioned Storage Facilities," in *SmartObjects*, ACM, Oct. 2016.

[45] B. Baron, M. Campista, P. Spathis, L. H. M. Costa, M. D. de Amorim, O. C. M. Duarte, G. Pujolle, and Y. Viniotis, "Virtualizing vehicular node resources: Feasibility study of virtual machine migration," *Vehicular Communications*, vol. 4, pp. 39–46, 2016.

[46] G. Pasaoglu, D. Fiorello, A. Martino, G. Scarcella, A. Alemanno, C. Zubaryeva, and C. Thiel, *Driving and Parking Patterns of European Car Drivers: A Mobility Survey*. Publications Office, 2012.

[47] R. A. Gorcitz, Y. Jarma, P. Spathis, M. D. de Amorim, R. Wakikawa, J. Whitbeck, V. Conan, and S. Fdida, "Vehicular carriers for big data transfers (poster)," in *IEEE Vehicular Networking Conference*, 2012.

[48] B. Baron, P. Spathis, H. Rivano, M. Dias De Amorim, Y. Viniotis, and J. Clarke, "Software-Defined Vehicular Backhaul," in *Wireless Days 2014*, IEEE and IFIP, Nov. 2014.

[49] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar, "Learning to predict driver route and destination intent," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, IEEE, 2006.

[50] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning.," in *AAAI*, 2008.

[51] L. Liao, D. J. Patterson, D. Fox, and H. Kautz, "Learning and inferring transportation routines," *Artificial Intelligence*, 2007.

[52] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," in *UbiComp 2006: Ubiquitous Computing*, Springer, 2006.

[53] R. Church and C. R. Velle, "The maximal covering location problem," *Papers in regional science*, 1974.

[54] I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "Alternative routes in road networks," *Journal of Experimental Algorithmics (JEA)*, 2013.

[55] E. Cascetta and A. Nuzzolo, "A modified logit route choice model overcoming path overlapping problems: specification and some calibration results for interurban networks,"

[56] H. J. Van Zuylen and L. G. Willumsen, "The most likely trip matrix estimated from traffic counts," *Transportation Research Part B: Methodological*, 1980.

[57] M. Pióro, P. Nilsson, E. Kubilinskas, and G. Fodor, "On efficient max-min fair routing algorithms," in *Computers and Communication, 2003.(ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, IEEE, 2003.

[58] M. H. Ammar and J. W. Wong, "The design of teletext broadcast cycles," *Performance Evaluation*, 1985.

[59] DGITM/DIT – SETRA – IGN, "Recensement de la circulation sur le réseau routier national en 2011." `http://tinyurl.com/otfbewv`.

[60] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The location of emergency service facilities," *Operations Research*, 1971.

[61] N. Megiddo, E. Zemel, and S. L. Hakimi, "The maximum coverage location problem," *SIAM Journal on Algebraic Discrete Methods*, 1983.

[62] A. Getis and J. K. Ord, "The analysis of spatial association by use of distance statistics," *Geographical analysis*, 1992.

[63] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *ACM Simutools*, 2009.

[64] N. Sarafijanovic-Djukic, M. Piorkowski, and M. Grossglauser, "Island hopping: Efficient mobility-assisted forwarding in partitioned networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1, pp. 226–235, IEEE, 2006.

[65] "Dublin bus gps sample data from dublin city council (insight project)." `http://dublinked.com/datastore/datasets/dataset-304.php`. Accessed: March 30, 2017.

[66] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, *et al.*, "Tinyos: An operating system for sensor networks," in *Ambient intelligence*, Springer, 2005.

[67] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, USENIX Association, 2005.

[68] K. Fall, "A delay-tolerant network architecture for challenged internets," in *ACM SIGCOMM*, 2003.

[69] "A conversation with jim gray," *ACM Queue*, vol. 1, pp. 8–17, June 2003.

[70] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1360–1369, IEEE, 2001.

[71] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network*, vol. 34. ACM, 2004.

[72] A. Vahdat, D. Becker, *et al.*, "Epidemic routing for partially connected ad hoc networks," tech. rep., Technical Report CS-200006, Duke University, 2000.

[73] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Elsevier Ad Hoc Networks*, vol. 1, no. 2, pp. 215–233, 2003.

[74] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, pp. 125–138, ACM, 2006.

[75] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, p. 6, 2009.

[76] D. J. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates, "Infostations: A new system model for data and messaging services," in *Vehicular Technology Conference, 1997, IEEE 47th*, vol. 2, pp. 969–973, IEEE, 1997.

[77] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way)," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pp. 233–244, ACM, 2003.

[78] I. Leontiadis and C. Mascolo, "Geopps: Geographical opportunistic routing for vehicular networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pp. 1–6, Ieee, 2007.

[79] V. N. Soares, J. J. Rodrigues, and F. Farahmand, "Geospray: A geographic routing protocol for vehicular delay-tolerant networks," *Information Fusion*, vol. 15, pp. 102–113, 2014.

[80] J. Burrell, T. Brooke, and R. Beckwith, "Vineyard computing: Sensor networks in agricultural production," *Pervasive Computing, IEEE*, vol. 3, no. 1, pp. 38–45, 2004.

[81] P. McDonald, D. Geraghty, I. Humphreys, S. Farrell, and V. Cahill, "Sensor network with delay tolerance (sendt)," in *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, Aug 2007.

[82] S. Lahde, M. Doering, W.-B. Pöttner, G. Lammert, and L. Wolf, "A practical analysis of communication characteristics for mobile and distributed pollution measurements on the road," *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1209–1218, 2007.

[83] W. Zhao, Y. Chen, M. Amma, M. Corner, B. Levine, and E. Zegura, "Capacity enhancement using throwboxes in dtns," in *IEEE MASS*, 2006.

[84] S. S. Chawathe, "Inter-vehicle data dissemination in sparse equipped traffic," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pp. 273–280, IEEE, 2006.

[85] M. Ibrahim, P. Nain, and I. Carreras, "Analysis of relay protocols for throwbox-equipped dtns," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009. 7th International Symposium on*, pp. 1–9, IEEE, 2009.

[86] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav, "Low-cost communication for rural internet kiosks using mechanical backhaul," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, pp. 334–345, ACM, 2006.

[87] M. Demmer and K. Fall, "Dtlsr: delay tolerant routing for developing regions," in *Proceedings of the 2007 workshop on Networked systems for developing regions*, p. 5, ACM, 2007.

[88] K. Chen and H. Shen, "Dtn-flow: inter-landmark data flow for high-throughput routing in dtns," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 212–226, 2015.

[89] M. Zarafshan-Araki and K.-W. Chin, "Trainnet: A transport system for delivering non real-time data," *Elsevier Computer Communications*, 2010.

[90] N. Banerjee, M. D. Corner, and B. N. Levine, "An energy-efficient architecture for dtn throwboxes," in *INFOCOM 2007. 26th IEEE international conference on computer communications. IEEE*, pp. 776–784, IEEE, 2007.

[91] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: Rethinking connectivity in developing nations," *IEEE Computer*, 2004.

[92] A. Rabagliati, "Wizzy digital courier–how it works," 2004.

[93] D. Waitzman, "A standard for the transmission of ip datagrams on avian carriers," RFC 1149, April 1990.

[94] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 252–259, ACM, 2005.

[95] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pp. 79–85, IEEE, 2007.

[96] B. Burns, O. Brock, and B. Levine, "Mv routing and capacity building in disruption tolerant networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 398–408 vol. 1, March 2005.

[97] J. Davis, A. H. Fagg, B. N. Levine, *et al.*, "Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks," in *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, pp. 141–148, IEEE, 2001.

[98] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *ACM Sigplan Notices*, vol. 37, no. 10, pp. 96–107, 2002.

[99] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 11, pp. 1576–1589, 2011.

[100] R. Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy, "Turning the postal system into a generic digital communication mechanism," in *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 159–166, ACM, 2004.

[101] B. Cho and I. Gupta, "Budget-constrained bulk data transfer via internet and shipping networks," in *Proceedings of the 8th ACM international conference on Autonomic computing*, pp. 71–80, ACM, 2011.

[102] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies. Proceedings IEEE*, vol. 2, pp. 1407–1418, IEEE, 2005.

[103] W. Zhao and M. H. Ammar, "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *Distributed Computing Systems, 2003. FTDCS 2003. Proceedings. The Ninth IEEE Workshop on Future Trends of*, pp. 308–314, IEEE, 2003.

[104] A. Mansy, M. Ammar, and E. Zegura, "Deficit round-robin based message ferry routing," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pp. 1–5, IEEE, 2011.

[105] Y. Tirta, Z. Li, Y.-H. Lu, and S. Bagchi, "Efficient collection of sensor data in remote fields using mobile collectors," in *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pp. 515–519, IEEE, 2004.

[106] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[107] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Replication routing in dtns: a resource allocation approach," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 596–609, 2010.

[108] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks.," in *IEEE INFOCOM*, 2006.

[109] J. Zhou, S. Roy, J. Li, R. Hu, and Y. Qian, "Minimizing the average delay of messages in pigeon networks," *Communications, IEEE Transactions on*, vol. 61, no. 8, pp. 3349–3361, 2013.

[110] O. Tekdas, V. Isler, J. H. Lim, and A. Terzis, "Using mobile robots to harvest data from sensor fields," *IEEE Wireless Communications*, vol. 16, no. 1, p. 22, 2009.

[111] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM MobiHoc*, 2004.

[112] M. M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pp. 37–48, ACM, 2006.

[113] K. C. Lee, M. Le, J. Harri, and M. Gerla, "Louvre: Landmark overlays for urban vehicular routing environments," in *IEEE VTC 2008-Fall*, 2008.

[114] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: an efficient routing in disruption-tolerant networks," in *ACM MobiHoc*, May 2009.

[115] N. Laoutaris, G. Smaragdakis, R. Stanojevic, P. Rodriguez, and R. Sundaram, "Delay-tolerant bulk data transfers on the internet," *IEEE/ACM Transactions on Networking*, vol. 21, pp. 1852–1865, Dec. 2013.

[116] B. Burns, O. Brock, and B. N. Levine, "Mora routing and capacity building in disruption-tolerant networks," *Ad hoc networks*, vol. 6, no. 4, pp. 600–620, 2008.

[117] M. Asadpour, K. A. Hummel, D. Giustiniano, and S. Draskovic, "Route or carry: Motion-driven packet forwarding in micro aerial vehicle networks," *IEEE Transactions on Mobile Computing*, 2016.

[118] B. Tan, J. Jose, X. Wu, and L. Ying, "A vehicular backbone network (vbn) with joint transportation-wireless capacity utilization," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*, pp. 549–556, IEEE, 2014.

[119] A. Keränen and J. Ott, "DTN over aerial carriers," in *ACM Mobicom workshop on challenged networks (CHANTS)*, (Beijing, China), Sept. 2009.