Modeling and Measuring Performance of Data Dissemination in Opportunistic Networks

THÈSE N^{O} 5448 (2012)

PRÉSENTÉE LE 21 SEPTEMBRE 2012

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 2

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Nikodin RISTANOVIĆ

acceptée sur proposition du jury:

Prof. B. Moret, président du jury
Prof. J.-Y. Le Boudec, directeur de thèse
Prof. A. Chaintreau, rapporteur
Prof. J.-P. Hubaux, rapporteur
Dr E. Yoneki, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

Abstract

In this thesis we focus on understanding, measuring and describing the performance of Opportunistic Networks (ONs) and their applications. An "opportunistic network" is a term introduced to describe a sparse, wireless, ad hoc network with highly mobile nodes. The opportunistic networking paradigm deviates from the traditional end-to-end connectivity concept: Forwarding is based on intermittent connectivity between mobile nodes (typically, users with wireless devices); complete routes between sources and destinations rarely exist. Due to this unique property of spontaneous link establishment, the challenges that exist in ONs are specific.

The unstructured nature of these networks makes it difficult to give any performance guarantees on data dissemination. For this reason, in *Part I* of this thesis we explore the dynamics that affect the performance of opportunistic networks. We choose a number of meaningful scenarios where our models and algorithms can be validated using large and credible data sets. We show that a drift and jump model that takes a spatial approach succeeds in capturing the impact of infrastructure and mobile-to-mobile exchanges on an opportunistic content update system. We describe the effects of these dynamics by using the age distribution of a dynamic piece of data (*i.e.,* information updates) as the performance measure. The model also succeeds in capturing a strong bias in user mobility and reveals the existence of regions, whose statistics play a critical role in the performance perceived in the network. We exploit these findings to design an application for greedy infrastructure placement, which relies on the model approximation for a large number of nodes.

Another great challenge of opportunistic networking lies in the fact that the bandwidth available on wireless links, coupled with ad hoc networking, failed to rival the capacity of backbones and to establish opportunistic networks as an alternative to infrastructure-based networks. For this reason, we never study ONs in an isolated context. Instead, we consider the applications that leverage interconnection between opportunistic networks and legacy networks and we study the benefits this synergy brings to both. Following this approach, we use a large operator-provided data set to show that opportunistic networks (based on Wi-Fi) are capable of

offloading a significant amount of traffic from 3G networks. At the same time, the offloading algorithms we propose reduce the amount of energy consumed by mobiles, while requiring Wi-Fi coverage that is several times smaller than in the case of real-time offloading. Again we confirm and reuse the fact that user mobility is biased towards certain regions of the network.

In *Part II* of this thesis, we treat another issue that is essential for the acceptance and evolution of opportunistic networks and their applications. Namely, we address the absence of experimental results that would support the findings of simulation based studies. Although the techniques such as contact-based simulations should intuitively be able to capture the performance of opportunistic applications, this intuition has little evidence in practice. For this reason, we design and deploy an experiment with real users who use an opportunistic Twitter application, in a way that allows them to maintain communication with legacy networks (*i.e.,* cellular networks, the Internet). The experiment gives us a unique insight into certain performance aspects that are typically hidden or misinterpreted when the usual evaluation techniques (such as simulation) are used. We show that, due to the commonly ignored factors (such as the limited transmission bandwidth), contact-based simulations significantly overestimate delivery ratio and obtain delays that are several times lower than those experimentally acquired. In addition to this, our results unanimously show that the common practice of assuming infinite cache sizes in simulation studies, leads to a misinterpretation of the effects of a backbone on an opportunistic network. Such simulations typically overestimate the performance of the opportunistic component, while underestimating the utility of the backbone. Given the discovered deficiencies of the contact-based simulations, we consider an alternative statistical treatment of contact traces that uses the weighted contact graph. We show that this approach offers a better interpretation of the impact of a backbone on an opportunistic network and results in a closer match when it comes to modeling certain aspects of performance (namely, delivery ratio).

Finally, the security requirements for the opportunistic applications that involve an interconnection with legacy networks are also highly specific. They cannot be fully addressed by the solutions proposed in the context of autonomous opportunistic (or ad hoc) networks, nor by the security frameworks used for securing the applications with continuous connectivity. Thus, in *Part III* of this thesis, we put together a security framework that fits the networks and applications that we target (*i.e.,* the opportunistic networks and applications with occasional Internet connectivity). We then focus on the impact of security print on network performance and design a scheme for the protection of optimal relaying capacity in an opportunistic multihop network. We fine-tune the parameters of our scheme by using a game-theoretic approach and we demonstrate the substantial performance gains provided by the scheme.

Keywords

Opportunistic networks, delay-tolerant networks, pocket switched networks, hybrid networks, performance modeling, system design, securing opportunistic communication, network measurement, validating simulation with measurements, optimization, algorithm design and analysis, offloading 3G networks, energy-efficient architecture.

Résumé

Dans cette thèse, nous nous concentrons sur la compréhension, la mesure et la description des performances des réseaux opportunistes (ROs) et de leurs applications. Le terme “réseau opportuniste” a été introduit pour décrire des réseaux sans fil ad hoc clairsemés avec des nœuds très mobiles. Le paradigme de réseau opportuniste s’écarte de la notion de connectivité de bout en bout traditionnelle. La transmission de données est basé sur la connectivité intermittente entre les nœuds mobiles (ce sont typiquement les utilisateurs avec des périphériques sans fil). Le chemin complet entre la source et la destination existe rarement. Grâce à cette propriété unique de l’établissement du lien spontané, les défis qui existent dans les ROs sont spécifiques.

Tout d’abord, la nature non structurée de ces réseaux rend difficile toute garantie sur la performance de la diffusion des données. Pour cette raison, dans la première partie de cette thèse, nous explorons les dynamiques qui affectent les performances des réseaux opportunistes. Nous examinons une sélection de scénarios significatifs, où nos modèles et algorithmes peuvent être validés en utilisant des jeux de données réalistes. Nous montrons qu’un modèle “drift and jump”, qui adopte une approche spatiale, réussit à capter les effets de la mobilité, de l’infrastructure et des échanges directs entre les nœuds, sur la distribution d’âge d’information dynamique, dans un système de mise à jour opportuniste. Nous montrons aussi que la mobilité des utilisateurs est biaisée en faveur de certaines régions, dont les statistiques jouent un rôle crucial dans la performance perçue dans le réseau. Nous appliquons ces resultats afin de concevoir un algorithm glouton de placement d’infrastructure, qui repose sur l’approximation du modèle pour un grand nombre de nœuds.

Un autre grand défi des réseaux opportunistes réside dans le fait que la bande passante disponible sur les liaisons sans fil, couplée avec les réseaux ad hoc, n’a pas réussi à rivaliser avec la capacité des backbones ni à établir des réseaux opportunistes comme une alternative aux réseaux infrastructurels. Pour cette raison, nous n’avons jamais étudie les ROs dans un contexte isolé. Au lieu de cela, nous considérons les applications qui bénéficient de l’interconnexion entre des réseaux opportunistes et les réseaux existants et nous étudions les avantages

que cette synergie apporte aux deux. Conformément à cette approche, nous utilisons les données fournies par un grand opérateur afin de montrer que les réseaux opportunistes (basés sur une connexion Wi-Fi) sont capables de décharger un montant significatif du trafic des réseaux 3G. Dans le même temps, les algorithmes de déchargement que nous proposons réduisent la quantité d'énergie consommée par les mobiles, tout en exigeant une couverture Wi-Fi plusieurs fois plus petite que la couverture nécessaire en cas de déchargement en temps réel. Encore une fois, nous confirmons et nous réutilisons le fait que la mobilité des utilisateurs est biaisée en faveur de certaines régions du réseau.

Dans la deuxième partie de cette thèse, nous traitons une autre question qui est essentielle pour l'acceptation et l'évolution des réseaux opportunistes et de leurs applications. Nous adressons le problème de l'absence de résultats expérimentaux qui appuient les conclusions des études par simulation. Les techniques comme la simulation reposant sur les contacts devraient être intuitivement capables de nous donner la performance des applications opportunistes. Cependant, cette intuition a peu de preuves dans la pratique. Pour cette raison, nous concevons une expérience avec les utilisateurs réels qui utilisent une application Twitter opportuniste, d'une manière qui leur permet de maintenir la communication avec les réseaux existants (c'est-à-dire réseaux cellulaires, Internet). L'expérience nous donne un aperçu unique sur certains aspects de la performance qui sont généralement cachés ou mal interprétés, lorsque les techniques classiques d'évaluation (telles que la simulation) sont utilisés. Nous montrons qu'en raison de facteurs souvent ignorés (comme la bande passante limitée), les simulations reposant sur des contacts surestiment considérablement le taux de transfert et sous-estiment le retard. De plus, nos résultats montrent à l'unanimité que la pratique courante dans les simulations de considérer des tailles de cache infinies, conduit à une interprétation erronée des effets du "backbone" sur un réseau opportuniste. Ces simulations en général surestiment la performance du composant opportuniste, tandis qu'elles sous-estiment l'utilité du "backbone". Au vu des lacunes constatées dans les simulations reposant sur des contacts, nous proposons d'utiliser une autre méthode de nature statistique reposant sur le graphe de contact pondéré. Nous montrons que cette approche offre une meilleure interprétation de l'impact d'un "backbone" sur un réseau opportuniste et entraine de bons resultats quand il s'agit de la modélisation de certains aspects de la performance du réseau (par exemple le taux de ransfert).

Enfin, les exigences de sécurité pour les applications opportunistes qui impliquent l'interconnexion avec les réseaux existants ne peuvent pas être satisfaites par les solutions proposées dans le contexte des réseaux opportunistes autonomes, ni par les solutions utilisés pour sécuriser les applications avec une connectivité continue. Ainsi, dans la troisième partie de cette

thèse, nous mettons en place une solution de sécurité qui convient aux réseaux et aux applications que nous visons. Nous nous concentrons par la suite sur l'impact de la sécurité sur la performance du réseau et nous concevons un système assurant la protection de la capacité de transmission optimale dans un réseau opportuniste multi-hop. Nous réglons les paramètres de notre système en utilisant une approche de la théorie des jeux et nous montrons que notre solution améliore la performance de maniére substantielle.

Mots clés

Réseaux opportunistes, delay-tolerant networks, pocket switched networks, réseaux hybrides, modélisation des performances, conception des systèmes, sécurisation de la communication opportuniste, mesures du réseau, validation des simulations avec des mesures, optimisation, conception et analyse des algorithmes, déchargement des réseaux 3G, architectures énergétiquement efficaces.

Acknowledgments

First and foremost, I would like to thank my thesis director, Professor Jean-Yves Le Boudec for his constant encouragement, guidance and support, both on a scientific and personal level. The time spent in his lab offered me an unmatchable exposure to high-quality research, teaching and work management. For all these reasons, I feel extremely privileged to have been given an opportunity to do my PhD thesis with him.

I would also like to thank Professors Augustin Chaintreau, Jean-Pierre Hubaux, Bernard Moret and Eiko Yoneki for accepting to be in my PhD jury and for taking time to evaluate this work and provide helpful comments.

I owe my gratitude to Professor Augustin Chaintreau, Dr. Vijay Erramilli, Dr. Laurent Massoulié and Dr. Pablo Rodriguez for offering me the opportunity to do an internship with Technicolor Paris Lab and Telefonica Research Lab in Barcelona.

I am much obliged to a number of former and current lab members for welcoming me to the LCA lab and for making me feel at home during all these years. I would especially like to thank to Dr. Dan-Cristian Tomozei for being such a great officemate. I would also like to thank him and Dr. Nicolas Gast for their relentless answers to my numerous linguistic questions.

In the course of my PhD I was very fortunate to collaborate with some brilliant researchers. I am particularly grateful to Dr. George Theodorakopoulos, whose impressive breadth of theoretical knowledge made working with him a truly enriching experience.

I also owe thanks to our secretaries (Danielle, Holly, Angela and Patricia) and our system administrators (especially to Marc-André Lüthy) for always being there to help with all the non-scientific tasks related to my PhD.

I am grateful to my friends at and outside of EPFL (especially to David, Mihailo, Lala, Che, Igor, Nebojsa, Darko) for all the good times we had together in the course of this PhD.

Finally, I am enormously grateful to my family (from Serbia to New Zealand) for all their love and support and I dedicate this thesis to them.

List of Abbreviations

CA	certification authority
CCDF	complementary cumulative distribution function
CDF	cumulative distribution function
CDR	call detail record
DoS	denial of service
DTN	delay tolerant network
EC-DSA	elliptic curve - digital signature algorithm
FIFO	first in, first out
HTTPS	hypertext transfer protocol secure
HU	home user
HUE	home user equivalent
IVC	inter-vehicle communication
LFU	least frequently used
LRU	least recently used
M2M	mobile-to-mobile
MANET	mobile ad hoc networks
ODE	ordinary differential equation
ON	opportunistic networks
PDE	partial differential equation
PKC	public key cryptography
PKI	public key infrastructure
PSN	pocket switched networks
RSA	Rivest-Shamir-Adleman algorithm
RSU	road-side unit

RT	real-time
RU	roaming user
UMP	user mobility profile
VC	vehicular communication

Contents

Introduction and Related Work

Chapter 1

Introduction

1.1 Motivation

A decade ago, mobile phones were voice-centric devices, capable of sending text messages. Today, they are powerful data-centric computers, equipped with the ever-evolving cellular data connection and multiple short-range wireless interfaces, such as Wi-Fi and Bluetooth. This enables mobile phones to establish direct communication among themselves, but also to act as gateways towards legacy networks.

The rapid evolution of mobile devices gave birth to a new paradigm of Opportunistic Networks (ON) that goes beyond the concept of Mobile Ad Hoc Networks (MANET). Opportunistic networks are based on intermittent connectivity between users with wireless devices. They are normally built around people, typically pedestrians. In this context, they are also referred to as Pocket Switched Networks (PSN). As users are typically limited in speed, and much slower than the data propagation over wired or wireless links, delay in such networks remains non-negligible. Thus, more generally, opportunistic networks fall under the Delay Tolerant Networking (DTN) space.

As the idea of opportunistic networking deviates from the traditional end-to-end connectivity concept, the challenges also differ. First, the unstructured nature of these networks makes it extremely difficult to put any performance guarantees on data dissemination in opportunistic networks. For this reason, a large part of this thesis is dedicated to modeling and measuring information propagation and to the amelioration of the evaluation techniques, with the goal of improving our understanding of data dissemination in opportunistic networks.

Second, the bandwidth available on wireless links, coupled with ad hoc networking, failed

to rival the capacity of backbones and to establish opportunistic networking as an alternative to infrastructure-based networks. Thus, we focus on designing innovative services that are based on interconnection with legacy networks. In other words, we do not study opportunistic networks in an isolated context. Instead, we consider the applications that leverage interconnection between the opportunistic networks and the backbone-based networks and we study the benefits that this synergy brings to both.

Finally, the security requirements for the networks and applications that involve an opportunistic component, as well as the interconnection with the legacy networks (*i.e.,* the Internet or cellular networks) are rather specific. They cannot be fully addressed by the security solutions proposed in the context of autonomous opportunistic networks, nor by the security frameworks used for securing networks and applications with continuous connectivity. Thus, we put together a security framework that fits the networks and applications that we target. We then focus on the performance of this security solution, in order to design a method for protection of the relaying capacity in the opportunistic (multi-hop) networks.

Although we concentrate on a few challenges, the complex interplay between closely related issues in opportunistic networking makes it difficult to ignore other key aspects. For example, it is impossible to consider data dissemination and collaboration without addressing incentives or user mobility. User mobility is particularly challenging. It can be beneficial, as it allows users to carry large amounts of data around the network. However, it also complicates the communication due to the instability of forwarding paths.

Similarly, it is impossible to design and implement an opportunistic application without considering the communication paradigms and related architectural aspects. Traditional communication paradigms such as client-server are not suitable for the opportunistic environment. Thus, other models, such as an event-based communication model and proximity-based group communication have to be considered in this context.

1.2 Dissertation Outline

This thesis is organized in three parts. In Part I, we characterize the dynamics that affect the performance of data dissemination in large scale opportunistic networks. In Chapter 3, we demonstrate that a continuous Markov chain model that takes a spatial approach allows us to describe how mobility, opportunistic exchanges and content inputs by arbitrary placed sources affect the age distribution of a dynamic piece of information in an opportunistic content-update system. We then use the fluid approximation of the model, which can be entirely characterized

by a system of ordinary differential equations (ODEs), for the design of an application for infrastructure dimensioning.

In Chapter 4, we use a much larger operator provided data set, with half a million users, to study an application of opportunistic networks with a greater reach. Specifically, we study their capacity to offload a part of the traffic from the congested 3G networks. To offload bulky, socially recommended contents from 3G networks we propose two algorithms that leverage prediction, delay-tolerance and the global view of the network available to mobile operators. Just like in Chapter 3, we exploit the fact that user mobility is biased towards a few regions of the network, whose statistics play a critical role in the performance seen by users. We perform extensive performance evaluation of both proposed algorithms and compare them with an existing real-time offloading solution. We show that our delay-tolerant algorithms reduce the amount of energy consumed by mobile phones, thus making them appealing to the users. At the same time the algorithms are interesting to the operators, as they leverage operators' clouds and offer load balancing between the orthogonal wireless technologies.

Chapter 5 constitutes Part II of this thesis. In this chapter, we go one step further and study the performance of an opportunistic network through a real application. The application extends a popular social service on the Internet (Twitter) to the space of intermittently connected opportunistic clients. We compare the measured performance with the results obtained by using a standard methodology for performance evaluation of opportunistic networks and applications. More precisely, we examine the gap between the performance of an opportunistic network obtained via the commonly used contact-based simulations and the performance acquired from a real deployment of such a network. This is an important problem, which is complex to study, because, although the trace-based simulations are omnipresent, live deployments of opportunistic networks and their performance measurements are virtually non-existent. For this reason, we deploy a testbed with our opportunistic Twitter application on the EPFL campus site. The setup and the three-week long experiment that we perform enable us to collect both the application data and the contact traces for 50 experiment participants. We then use the collected data sets for the comparison between the measured application performance and the results of the contact-based simulations.

In addition to this, we use our experiment to study the effects of a backbone on an opportunistic network. This is possible because the implemented opportunistic Twitter application uses a backbone to help forward the tweets created by the experiment participants and to retrieve tweets from the Twitter web site. By comparing again the values obtained from the experiment with the results of the contact-based simulations, we find that the simulations fail to

capture the effects of adding a backbone to an opportunistic network. The simulations typically overestimate the performance of the opportunistic component and underestimate the utility of a backbone. We analyze the assumptions used in the case of contact-based simulations and offer an explanation for this behavior. Finally, in the last part of Chapter 5 we propose an alternative statistical treatment of contact traces (as opposed to trace driven simulation) that uses the weighted contact graph. We show that this approach offers better interpretation of the impact of a backbone on the opportunistic network and results in a closer match when it comes to modeling certain aspects of network performance (namely delivery ratio).

Chapters 6 and 7 constitute Part III of this thesis. In Chapter 6 we concentrate on the security aspects of the opportunistic Twitter application used in the experiment (and other similar applications with occasional Internet connectivity). We explain the differences in security requirements, between the opportunistic application clients with intermittent connectivity and the traditional (always connected) mobile clients, and we propose an architecture for securing the former. The proposed solution contains certain elements of several security frameworks designed for different environments, such as the PKI building blocks used to secure vehicular communication and the OAuth authorization used for authentication of the always-connected clients.

As the performance of opportunistic networks and applications represents the key focus of this thesis, in Chapter 7 we revisit this topic. More precisely, we concentrate on the performance of the opportunistic security framework designed in Chapter 6. We show how it affects the performance of opportunistic applications, by observing the relaying capacity of mobile nodes. Hence, we design a scheme that complements the security framework proposed in Chapter 6 and protects the optimal relaying capacity in an opportunistic network. We fine-tune the parameters of our scheme by using the min-max approach and we demonstrate the substantial performance gains provided by the scheme.

Finally, we conclude this thesis in Chapter 8 with a summary of the main findings and a discussion of possible directions for future work.

1.3 Contributions

The following is the list of the main contributions of this thesis.

• We show that a drift and jump model that takes a spatial approach allows us to characterize the age distribution of a dynamic piece of information in an opportunistic content update system, with arbitrary mobility, contact rates and locations of input sources. Using a 30-day

trace with 500 taxicabs in San Francisco area, we show that, in addition to the model, its fluid approximation fits the data well. This allows us to use it as a fast simulation tool in the cases when traces are not available, or to perform a what-if analysis, or when the number of mobile nodes is very large. We propose an infrastructure dimensioning application that uses the ordinary differential equations (ODEs) stemming from the fluid approximation.

• We quantify the potential for offloading 3G data networks through the usage of opportunistic networking. We design two algorithms for delay-tolerant offloading of large, socially recommended content from 3G networks, and we show their advantages over the real-time offloading solution currently deployed by some mobile operators. We perform a comprehensive evaluation of the algorithms by using a large, operator provided data set, with more than half a million users. We find that both algorithms succeed in offloading a significant amount of traffic, with a positive effect on user battery lifetime. We show that the Wi-Fi coverage needed to offload traffic is reduced very quickly (by a factor of 3 to 4) when some delay is tolerated. Finally, we show that both algorithms deliver content with the lowest delays during the peak hours, when offloading is most needed, which means that opportunistic transfers can naturally complement the energy-costly 3G data downloads.

• We address the problem of the missing evidence that the results of commonly-used contact-based simulations accurately reflect performance of opportunistic networks and we find significant gaps between the two. For this purpose, we design and implement a testbed with a real application and real users, which allows us to collect application data in addition to the contact traces and compare measured performance to the results of the contact-based simulations. We show that although the contact-based simulations succeed in capturing the relative effects of different system parameters, there exist significant discrepancies between the values obtained through simulation and those obtained from the experiment. We show that, due to some commonly ignored factors (such as the limited contact durations, finite transmission bandwidth, technology limitations, etc.) [1, 2], contact-based simulations significantly overestimate delivery ratio, and the acquired delays are 2-3 times lower than the experimentally obtained delays.

• Additionally, the results of our three week-long experiment, with 50 users and a range of cache sizes and caching strategies, unanimously confirm that the common practice of assuming infinite cache sizes in simulation studies [3] leads to misinterpretation of the effects of a backbone on an opportunistic network. Our results show that the conclusions about the utility of a backbone [3] tend to be largely pessimistic. This is an important finding, as it could direct more attention towards hybrid networks that include both an opportunistic and an infrastruc-

tural component

• We show that a statistical treatment of the contact trace, by using the weighted contact graph, offers a better prediction of certain performance aspects (namely delivery ratio) than the trace driven simulation. We expose a strong dependency between a user centrality measure in this graph and the perceived delivery ratio, and we fit a simple curve to this dependency. This allows one to predict users' delivery ratios based on the contact trace. We show that this dependency persists when a backbone is added to the network, which means that it can be used to estimate the effects of adding infrastructure to an opportunistic network.

• From the application aspect, our experiment results in the implementation of a full-fledged opportunistic application for Windows Mobile and Android platforms. The application leverages intermittent Internet connectivity and multi-hop forwarding, enabling mobile users to use the popular Twitter application in the opportunistic fashion even when the Internet connectivity is not available (for example when in roaming).

• We propose a full security framework adapted to the specific requirements of the family of opportunistic applications that we target (*i.e.,* the opportunistic applications that synchronize with existing web services). The framework combines elements of the novel security solutions (such as OAuth) used to authenticate mobile clients with continuous Internet connectivity and the PKI based architecture proposed in the context of vehicular networks.

• Finally, we address the performance of the proposed security framework. In order to protect the optimal relaying capacity in an opportunistic network we design an adaptive scheme that can be easily integrated with the proposed security framework. Using extensive simulations, we show that the scheme resists DoS attacks and yields a significant performance increase, irrespective of the number of adversaries in the network.

Chapter 2

Overview of Related Work

2.1 Evolution of Opportunistic Networking Paradigm

The idea of Delay-Tolerant Networking (DTN) was initially inspired by the interplanetary communication or the interplanetary Internet [4]. The traditional Internet protocols are not applicable in such an environment, due to high delay (latency), limited resources and only intermittent connectivity (planet rotation).

The increasing availability of wireless networks and the proliferation of wireless-equipped devices made the DTN paradigm attractive in other communication contexts [5]. Above all, the usage of delay-tolerant networking started to be considered in scenarios with mobile users or vehicles, capable of direct wireless communication [1, 6]. In such environments DTNs are often referred to as Opportunistic Networks (ONs), as wireless transmission opportunities normally arise in opportunistic (spontaneous) fashion. In the opportunistic networks, end-to-end routes can rarely be established. This means that opportunistic networking goes beyond the concept of Mobile Ad Hoc Networks (MANETs), which usually focus on end-to-end routing between mobile nodes.

Opportunistic networks are typically considered in environments with intermittently connected wireless nodes, where standard Internet protocols can not be applied or would provide poor performance. For example, ONs have been proposed as means of bridging between isolated rural areas [7, 8], wildlife monitoring [9, 10] and networking using buses that follow predictable routes [11].

2.1.1 Pocket Switched Networks (PSNs)

A subset of papers in the area of delay-tolerant networks focus exclusively on opportunistic networks formed by small human-carried devices. In this context, the ONs are often referred to as Pocket Switched Networks (PSN) [12, 13]. The specificity of these networks is that data transmissions occur between mobile nodes with seemingly random mobility.

PSNs can enable data exchanges with moderate delays within small "connectivity islands" [12]. However, outside these islands connectivity becomes a major problem. For this reason our approach in this thesis is to always take into account the existing network infrastructure, which can serve as a backbone or a bridge between connectivity islands. In other words, we concentrate on the scenarios where opportunistic networks can complement the existing network infrastructure, like in the case of 3G traffic offloading [14] or inexpensive synchronization of data services in roaming [15]. Such scenarios allow opportunistic networks to provide added value to infrastructure networks, instead of acting as an alternative. We believe this makes them more meaningful and attractive from the user point of view.

Apart from this important difference, models, algorithms and applications proposed in this thesis are close to the PSN context. For example, our solution for the cheap synchronization of social mobile applications in roaming relies on mobile users (*i.e.,* cellular subscribers in their home networks that can be found in proximity). The 3G offloading algorithm that we propose and evaluate is also based on the available opportunistic (mobile-to-mobile) bandwidth. Our model of age in an opportunistic content update system [16] is evaluated using mobile nodes that do not follow predefined mobility patterns (unlike buses in [6]).

2.1.2 The Role of Infrastructure in Opportunistic Networking

After the early efforts to design opportunistic networks that would be formed entirely by (human carried) mobile devices [17, 18, 19, 20, 21], a part of the research community has begun looking into the ways to reduce delays and improve delivery ratios, by adding certain amount of infrastructure to these networks [22, 3, 23]. The reason for this is the discovery that human contact processes exhibit heavy-tailed inter-contact distributions [1, 24]. The pronounced heavy-tail makes it difficult for any forwarding algorithm that relies only on multiple message copies, or on encounters with destinations, to deliver content within a reasonable time [1].

The initial studies of the effects of an infrastructure (a network backbone) on the opportunistic network performance relied exclusively on contact traces [22, 2, 3]. In [22], the authors consider how infrastructure can be used to design simpler and more efficient (in terms of delay

and number of hops) opportunistic forwarding algorithms. In [3] the authors perform extensive simulations using Bluetooth contacts in order to quantify the effects of opportunistic and backbone components on a DTN. They conclude that backbone brings *only marginal improvements to opportunistic communication*.

The UMass DieselNet testbed addressed a similar topic, but the Wi-Fi equipped buses exchanged traffic (artificially obtained from the Poisson distribution) [25, 23, 26]. This time, *much higher utility of the backbone component is observed* [25]. In order to find the origins of this important discrepancy, we dedicate an important part of this thesis to the comparison between the network performance obtained from experiments and the performance acquired from contact-based simulations. We reveal that much of the discrepancy in the observed backbone-induced improvement, comes from a common assumption in contact-based simulations, about the infinite cache sizes in mobile nodes.

2.2 Data Forwarding in Opportunistic Networks

One of the principal challenges of opportunistic networking is forwarding in the networks where end-to-end routes are rarely available and where connectivity between mobile nodes is not a priory known at any given time. Although we are not directly interested in the design of forwarding algorithms, a short overview of the existing approaches to forwarding in ONs can help put in context the choices we make in this thesis.

Most proposals that address forwarding in opportunistic networks require certain knowledge of network configuration. In a large number of cases the knowledge of connectivity schedule is required. In [6], historical data and lists of previous intermediaries are used to prioritize the schedule of packets transmitted to other peers and the schedule of packets to be dropped. In [18] forwarding decisions are made based on the topology information, which is flooded to all nodes inside the link-state packets. In [19], the authors propose PROPHET, a probabilistic routing protocol that makes forwarding decisions based on the computed delivery predictability of intermediate nodes. Leguay et al. addresses the forwarding problem with an algorithm that is based on the use of a high-dimensional Euclidean space, constructed upon nodes' mobility patterns. Their algorithm is based on the frequency of nodes' visits to each possible location in the network. In [27], the authors use communities obtained from a social graph to propose a forwarding algorithm based on social relations. Ioannidis et al. use so-called "weak ties" (*i.e.,* relationships with people outside the narrow social circles) to improve the dissemination of content updates over a mobile social network [28]. The main problem

of all these proposals is that they necessitate the knowledge about the network, which is often unavailable in a self organizing network.

2.2.1 Epidemic Data Forwarding

A special group of forwarding algorithms consists of approaches that rely on epidemic message replication. The major advantage of this class of forwarding algorithms is that it can operate with very little or no prior information about network organization. Most of these proposals assume that a node is equally likely to contact any other node at each time step. Nevertheless, it has been recently shown that similar performance can be attained when nodes contact each other, according to some general static graph topology [29, 30].

In addition to being simple and scalable, epidemic procedures were shown to be efficient with respect to their deterministic counterparts and robust in the face of topological changes [31]. Due to the fact that epidemic algorithms do not require coordinated collaboration among nodes, they have been proposed for routing in delay-tolerant networks, where topology is not a priori known [21]. Even the algorithms that can not be classified as purely epidemic, rely on epidemic algorithms as a primitive (usually flooding), which is then further improved using additional information and heuristics to decide which packets to transmit (*e.g.,* [6]).

The epidemic routing is particularly important from the viewpoint of the work done in this thesis, as we study an opportunistic content update system based on epidemic dissemination in Chapter 3. Closest to our work is [32], which studies different epidemic strategies for updates between nodes that are intermittently connected, and focuses on optimal control. However there are some important differences: first, we assume a more general model, where nodes move between classes and contact each other and the infrastructure with rates that depend on the classes. Though our model does not include cost, it allows to truly study the influence of mobility and geographical constraint on the performance of epidemic algorithm.

Also close to our work on epidemic dissemination (gossip) is the aforementioned [23], which compares delivery latency of meshes, base stations and mobile relays in opportunistic systems. Similarities are in the use of a multi-class model for spatial aspects, and in the use of differential equations. Our goals are significantly different, though. First, we want to characterize the complete latency distribution over all nodes and classes, rather than the dissemination of a single piece of content. To put it differently, and leaving aside the class attribute of a node, in [23] and [33], the state of one node is a single bit (infected or not) whereas in our case it is a nonnegative real number (the age of the node's content). Note that the age cannot be deduced

from the time since the last infection, since it depends on when the content was originally emitted by a base station. Thus we have a completely new way to evaluate the freshness of disseminated information.

2.3 Modeling Human Mobility

In any opportunistic content dissemination system that relies on transmissions between hand-held devices, human mobility has an important place. For this reason, a number of research efforts directed towards measuring and understanding human mobility (and related to opportunistic networking) can be found in the literature [34, 35, 12]. In terms of its impact on the pace of data dissemination, mobility has different and often conflicting roles. On one hand it can increase the available bandwidth, due to large amounts of data that can be carried by opportunistic nodes [36, 37]. On the other hand, mobility is the principle cause of disconnections between nodes in opportunistic networks.

The existing models of human mobility, such as Levy-walk model, are essentially stochastic, which means that they perceive human mobility as a random process [38, 39]. However, several recent studies have shown that contrary to the common beliefs about human mobility, humans follow repetitive and reproducible patterns [40, 41]. In [42] Song et al. measure entropy of users' trajectories and find 93% predictability in user mobility across the whole user base. Moreover they find that this predictability varies little across users and does not depend on distances covered by users on regular basis. Although the authors show the existence of predictability (by means of measuring entropy), they do not offer algorithms that would help us use it.

We go on step further and we design and evaluate an algorithm that allows us to extract and leverage predictability of user mobility (*e.g.,* for energy efficient offloading of 3G data). In addition to predictability in human mobility, we also detect (in multiple data sets) a strong mobility bias towards certain regions of the network. We show how this bias can be used to boost the performance of several applications that we consider (*i.e.,* citywide data dissemination, orthogonal 3G data offloading, etc.).

2.4 Power Efficiency Aspects of Opportunistic Networking

In line with our efforts to identify the added value that opportunistic networking can bring to the existing legacy networks, we seek to quantify the energy saving, which is achievable by

cellular subscribers when they use the available opportunistic bandwidth.

Energy optimization in wireless networks is a problem that draws a lot of attention in research community. A part of the community effort is directed towards the energy saving on the infrastructure side, namely, in Wi-Fi access points and in cellular base stations. In [43] the authors consider minimizing the number of base station sites. In [44, 45] switching off certain sites during the periods when they are under-utilized is proposed. This is typically achieved by re-arranging the user-cell associations.

Mobile users are much more concerned about the limited battery lifetime of their mobile devices and, thus, interested in solutions that can extend the battery duration. Comprehensive measurement studies of energy consumed by wireless smartphone interfaces are performed in [46, 47, 48]. They all show a strong impact of wireless interfaces on the battery consumption. In particular, they stress the high cost of 3G transmissions. Nonetheless, they also show that Wi-Fi scanning and idle state have rather high power consumptions, which means that continuous Wi-Fi discovery quickly drains the phone battery.

Due to this high energy print of wireless interfaces present on mobile devices, a vast body of work proposes different techniques that would help us use them in a more economical way. The proposals usually exploit the diversity of available wireless interfaces and mobility. They aim at improving energy efficiency, but also download speeds. In [49], the authors propose collaborative downloading as means of increasing download speeds and battery life. In [50], policies for switching between multiple interfaces are proposed, with the goal to increase battery lifetime. Namely, the authors propose switching between Wi-Fi and low-power Bluetooth radio, during data transmissions. Ananthanarayanan et al. [51] try to improve the energy efficiency of Wi-Fi by replacing Wi-Fi scanning with Bluetooth scanning. Unlike these efforts, we aim at estimating the energy saving achievable by cellular subscribers that comes from the use of opportunistic bandwidth.

Many existing proposals (including ours) that target a more efficient content delivery via wireless necessitate the wakeups of one or more wireless interfaces (that are asleep for the power efficiency reasons). This often requires certain modifications on the side of infrastructure. The exception is the work by Wu et al. [52]. They use cellular footprints to wake up the Wi-Fi interfaces on smartphones when in proximity of Wi-Fi APs. In [53], Agarwal et al. propose the use of a modified VoIP gateway that would turn on Wi-Fi whenever a VoIP call arrives. Closer to our work is the proposal by Shih et al. [54], who use a separate paging infrastructure to wake up Wi-Fi.

2.5 Securing Opportunistic Communication

Many previous works addressed the problem of security in opportunistic and vehicular networks. We provide an overview of both groups of proposals, as they are similar with many respects and they both have some common ground with the security solutions that we propose in Chapters 6 and 7 of this thesis.

Haggle project addressed the problem of security in autonomous opportunistic networks [55, 56, 57], which resulted in a number of security mechanisms that are built in the security manager of the opportunistic Haggle nodes [58]. In [55], the authors analyze the impact of denial of service attacks on epidemic forwarding protocols, without offering solutions to the problem. We address this problem in Chapter 7 of this thesis, where we evaluate our scheme for the reduction of security print. The main difference between Haggle security and the security solution we propose in Chapter 6 is that the former is designed for a completely autonomous opportunistic network. We, on the other hand, propose a security framework for the family of hybrid applications that synchronize with the Internet (*i.e.,* that resamble the opportunistic Twitter application described in Chapter 5).

A large number of studies addressed the problem of node authentication in an environment without a trusted authority, where node security credentials are unknown or unobtainable. Solis et al. [59, 60, 61] propose a method for establishing an initial security context using social contact information. They then relax authentication requirements in order to reduce security overhead. In [62], the system of invitations is used to grow an opportunistic network in a trusted way. In [63], the authors propose a fully self-organized public-key management system that allows users to perform authentication without any centralized services (*i.e.,* without any trusted authority). In [64], the authors present a method for key establishment over a radio link in peer-to-peer networks, based on the modified DH key agreement protocol, resistent to man-in-the-middle attack. Finally, the work by Asokan et al. [65] is closer to our work, as it represents a step towards a network security that is bootstrapped using the existing large-scale security infrastructure.

Unlike the efforts to secure opportunistic networks, the major efforts to secure vehicular communication assume the existence of a trusted authority (which is why they served as direct inspiration for a part of the security framework that we propose in Chapter 6). This assumption can be found in three major projects related to vehicular security, namely: the NoW project [66], the IEEE 1609.2 working group [67], and the SeVeCom project [68]. They all rely on a Certification Authority (CA) and public key cryptography to protect vehicular communi-

cation, *i.e.,* to provide message authentication and integrity.

In addition to these large projects, a number of works outlined challenges [69], described attacks [70], and offered solutions [71, 72, 73] in the field of vehicular networks security. Some of them propose alternatives to the use of public key cryptography for node authentication. In [72], symmetric key cryptography complements the public key operations, while in [73] group signatures are used.

The solutions proposed in [72] and [73] are also important from the aspect of our work on security print reduction, presented in Chapter 7. In [72], one of the driving forces behind the introduction of symmetric key cryptography is the reduction of security overhead. In [73] context-agnostic overhead reduction schemes are proposed. In [74, 75], the authors propose context-specific strategies for security print reduction. The investigation of the vehicular communications security overhead and its effect on system/application performance is extended in [76]. These works are complementary to the security print reduction scheme that we propose and their joint investigation with our scheme would be an interesting point for future work.

Part I

Modeling Data Dissemination in Opportunistic Networks

Chapter 3

Epidemic Content Dissemination: Model and Applications

Simple randomized procedures (also referred to as epidemic or gossiping procedures) have already been used in computer networks for delivering updates. More precisely, they were introduced to maintain consistency of a distributed database [77], offering an alternative to complex deterministic algorithms. Recently, the same epidemic principle was proposed in the context of data forwarding in an opportunistic network, where links between nodes are intermittent [21]. However, as nodes (users) move at moderate speeds (in comparison to the speed of wired/wireless medium) the delay of the epidemic forwarding in an opportunistic network is a priori non-negligible.

In this chapter, we consider the use of epidemic (gossiping) procedures in an opportunistic network, where mobile users receive occasional updates through source inputs (from a set of base stations). Our goal is to determine if gossiping allows for recent updates to be efficiently maintained, *i.e.,* delivered with low delay to a large number of users.

Applying the concept of epidemic dissemination to an opportunistic network with highly mobile users is a challenging problem. The lack of structure in these networks makes the modeling of the important dynamics that affect their performance a difficult task. To characterize the evolution of age (of the latest available update) in an opportunistic content update system, we use a spatial drift and jump model. The model takes into account mobility, source inputs and direct mobile-to-mobile (M2M) exchanges. Using a large dataset collected by San Francisco taxicabs, we show that this approach succeeds in capturing the age evolution with good accuracy. We then use the mean-field approximation of the model, introduced in [16], to design

an application for greedy infrastructure placement that maximizes the system performance for the given resources.

The content update system that we consider in this chapter utilizes mobile users, for the opportunistic dissemination of a single piece of content (that is of interest to all users). The content is constantly updated at a source and injected in the network via one or more base stations. In addition to these source inputs, the content propagates in the network as a result of opportunistic contacts between mobile users. Upon each contact between two mobile users the one with the most up-to-date copy forwards it to the other, following the epidemic principle.

The metric we are interested in is the *age* of the latest copy available to each user, or more precisely, the distribution of ages over all participating users. It is important to note that this age represents the delay between the broadcast of the content by a base station and its reception by a mobile node. Thus, the measure that we characterize in this chapter differs from the majority of the epidemiological (infection) models, which are typically interested in the time elapsed since the last infection of a node or the spread of a single piece of content.

To account for the fact that contacts between mobile nodes occur as a function of their location (rather than uniformly), we introduce the notion of spatial *classes*. A class that a mobile user belongs to at a given moment in time represents a part of his state description in the model that we use. A user's contacts, with base stations and other mobile users, occur at rates that depend on the current classes of these users. As the mobility of mobile users is expressed through the change of classes, our goal is not only to capture the age distribution over the observed user population, but to do this for each of the spatial classes.

The main contributions of this chapter can be summarized as follows. We show that a drift and jump model that takes a spatial approach allows us to characterize the age distribution of a dynamic piece of content in an opportunistic content update system. It successfully accounts for arbitrary mobility, contact rates and the locations of input sources. Using a 30-day trace with 500 taxicabs in San Francisco area, we show that, in addition to the model, its fluid approximation fits the data well. This allows us to use it as a fast simulation tool in the cases when traces are not available, or to perform a what-if analysis, or when the number of mobile nodes is very large. We propose an infrastructure dimensioning application that uses the ordinary differential equations (ODEs) that stem from the fluid approximation.

The rest of this chapter is organized as follows. After presenting the related work in the next section, we describe the model in more detail in Section 3.2. In Section 3.3, we state the main results derived in [16], related to the mean-field approximation of the model. These results and the discrete event simulation (stemming from the model) are validated by using a large data set

in Section 3.4. Finally, in Section 3.5, we describe an application that solves the problem of infrastructure placement using the equations obtained from the model.

3.1 Related Work

Gossip protocols or epidemic algorithms were used in the past to maintain mutual consistency among multiple database sites [77], for reliable dissemination to a large group [78], or for peer-to-peer live-streaming [79]. In addition to being simple and scalable, these procedures were shown to be efficient with respect to their deterministic counterparts and robust in the face of topological changes [31]. Most of these works assume that a node is equally likely to contact any other node at each time step. It was recently shown that similar performance can be attained when nodes contact each other, according to some general static graph topology [29, 30].

Because epidemic algorithms usually assume that nodes collaborate in an uncoordinated manner, they have also been proposed for routing in ad hoc or delay-tolerant networks where topology is not known *a priori* [21]. Most of the routing protocols proposed in opportunistic delay-tolerant networks rely on epidemic algorithms as a primitive (usually flooding), which is then further improved using additional information and heuristics to decide which packets to transmit (see, *e.g.,* [6]). The main difference with the previous works mentioned above is that messages between nodes are not exchanged randomly or in a static set of neighbors, but they rather follow contacts created by node mobility.

Close to our work is a study of different epidemic strategies for updates between nodes that are intermittently connected with focus on optimal control [32]. However there are important differences: First, we assume a more general model, where nodes move between classes and contact each other and a base node with rates that depend on the class. Although our model does not include cost, it allows us to truly study the influence of mobility and geographical constraints on the performance of epidemic algorithm. Second, we prove convergence to a mean field regime, whereas [32] mentions it as a plausible assumption. Third, we completely characterize the mean field regime by partial differential equations (PDEs), which allows us to both obtain efficient solution methods and derive analytical conclusions. In particular, we show that the dynamics of this system follow linear multidimensional ordinary differential equations (ODEs) when focusing on low and high age, which gives us new insight into the impact of base stations and opportunistic node contacts.

Also close to our work is a comparison of delivery latency of mesh, base stations and mobile relays in opportunistic systems [23]. Similarities are in the use of a multi-class model for spatial

aspects, and in the use of differential equations. Our results are significantly different, though. First, because we focus on opportunistic content updates, we want to characterize the complete distribution of latency among nodes and classes, rather than the dissemination of a single piece of content. To put it differently, and leaving aside the class attribute of a node, in [23] and [33], the state of one node is a single bit (infected or not), whereas in our case it is a non-negative real number (the age of the node's content). Note that the age cannot be inferred from the time since last infection, as it depends on when the content was originally emitted by a base station. Hence, we have a completely new way of evaluating the freshness of disseminated information. Showing convergence to a mean field regime in our case is entirely new (and non-trivial), whereas convergence to a mean field regime in the case of one bit of information per node as in [23] follows, for example, from [80]. Also note that one can derive the extent of infection from the age distribution, so, in some sense, our model generalizes the model in [23] (but note that [23] focuses on dimensioning rules that are not directly addressed in this paper).

We believe that our complete characterization of the age of gossip for a large system significantly complements previous works. It can be used as a building block to address future issues of cost efficiency when mobility plays an important role.

3.2 Specifications and Model

3.2.1 A Multiclass Approach

We assume that mobile users (*i.e.,* users of the opportunistic content update system) are distributed in a finite number of classes. A user may belong to only one class at a time. Its class may change with time. As different classes represent different locations (regions), the mobility of users is modeled through the change of classes. We assume that users in the same class are statistically equivalent, *i.e.,* two different users, in the same location (captured via their class), behave statistically the same with respect to the evolution of their information age.

We further assume that a collection of N users move and receive updates according to the following three dynamics:

• **Mobility:** There exists a finite collection $\mathcal{C} \subset \mathbb{N}$ of C classes, and each mobile user belongs to a only one class at any given time. We call $\rho_{c,c'}$ the rate of movements (transitions) from class c to class c' per time unit.

• **Source Emission:** At any time a user can receive updated content directly from the source (through one of the fixed base stations). This happens at rate μ_c for users that are in a class c.

• **Opportunistic Contacts:** A mobile user may meet opportunistically with other users in the same or other classes. In this case we assume that the user with the most recent information transmits it to the other. We define the parameters η_c, $c \in \mathcal{C}$ such that, whenever a pair of users both are in class c, they meet at a rate $\left(\frac{2\eta_c}{N-1}\right)$. This implies that the total contact rate in one class is $\frac{N_c(t)(N_c(t)-1)}{N-1}\eta_c$ where $N_c(t)$ is the number of mobile users currently in class c.

We also allow for opportunistic contacts among users in different classes. This applies to cases where classes represent different types of users in the same location, or, as in Section 3.4, to contacts across class boundaries, when classes represent neighboring subareas. We define $\beta_{\{c,c'\}}$, for $c \neq c'$ such that two users belonging to classes $c \neq c'$ meet with a rate $\left(\frac{2\beta_{\{c,c'\}}}{N-1}\right)$.

Note that a class may have no infrastructure (*i.e.,* $\mu_c = 0$). In this case, the updates can only come from users that visit different classes. Similarly a class may represent an inactive state (*i.e.,* $\mu_c = \eta_c = 0$) where users are not likely to meet at all.

Example 1 (Homogeneous network) There is $C = 1$ class. This is the simplest, but as we show in Section 3.4.2, not a realistic model. All mobile users are statistically the same and they are equally likely to meet with the information source at any time (at rate μ), as well as with each other (with rate η).

Example 2 (Classes as geographical regions) We can map a more realistic scenario to classes as follows. We divide a geographical area of interest to classes, where each class represents a sub-area. In some classes there are one or more information (content) sources. In such a class μ_c is the aggregate rate of injection of new updates at these sources, *i.e.,* the aggregate contact rate of mobile users with the sources. In Section 3.4, we explain how to measure μ_c. In other classes, where there are no sources, $\mu_c = 0$. We also introduce an extra class (class 16 in Section 3.4) to account for the mobile users that leave the area (classes) of interest.

We show in 3.4.2 that classes do matter, in the sense that a model with just one or two classes gives a poor fit to trace results, whereas the one with more classes results in a good fit.

Note that we assume in the model that the total number of mobile users N is constant. Nonetheless, as explained in Example 2 above, we can account for a variable number of users by introducing an extra class, to represent mobile users that are not present in the area of interest. Thus, with our model, N is in fact an upper bound on the number of users in the area of interest.

Metric We are interested in the age distribution at any time and in any class. We are interested in the following quantities.

• $u_c^N(t)$ is the fraction of users in class c at time t.

• $F_c^N(z,t)$ is the fraction of users at time t that are in class c and whose latest update (obtained from the sources or by gossiping) has age $\leq z$. Note that we have for any $t \geq 0$, $0 \leq F_c^N(z,t) \leq u_c^N(t)$, and $F_c^N(0,t) = 0$, $F_c^N(\infty,t) = u_c^N(t)$.

3.2.2 Drift and Jump Model

The evolution of the system above is captured in continuous time via a drift and jump process. The state of the system at time t is $(\vec{X}^N(t), \vec{c}^N(t)) = \left((X_n^N(t))_{n=1}^N, (c_n^N(t))_{n=1}^N\right)$, with:

$X_n^N(t)$: age of the most recent information update held by user (node) n.
$c_n^N(t)$: current class of user n.

The dynamics that affect ages are essentially characterized by:

• If users m, n meet at time t then $X_m^N(t) := X_n^N(t) := \min(X_m^N(t^-), X_n^N(t^-)$.

• If a user m meets a base station at time t then $X_m^N(t) = 0$.

• The age of a user increases at rate 1 in an interval where this user does not meet any other user(s) nor base station(s).

We now formally describes all details of our model.

Evolution of Users Between Classes

Let $\{\ K_{n,c,c'}\ \mid n \in N, c \in \mathcal{C}, c' \in \mathcal{C}, c \neq c'\ \}$ be $N \times C \times (C-1)$ independent Poisson processes such that $K_{n,c,c'}$ has a rate $\rho_{c,c'}$. Each point of this process denotes a possible transition from the class c to the class c' for a user n (the transition always exist, but it has no effect unless the user n is currently in state c). Thus

$$d\vec{c}^N = \sum_{n \in N} \sum_{c \in \mathcal{C}, c' \in \mathcal{C}, c' \neq c} (c' - c) \cdot 1_{\{c_n^N = c\}} \cdot \vec{e}_n dK_{n,c,c'}\,,$$

where $\vec{e}_m$ is the $N \times 1$ vector with 0 at all components except the mth which is equal to 1. We can rewrite the fraction of users in class c, u_c for any N and any time t as:

$$u_c^N(t) = \frac{1}{N} \sum_{n=1}^{N} 1_{\{c_n^N = c\}}\,.$$

The process $\left\{ \ (u_c^N(t))_{c\in\mathcal{C}} \ \middle| \ t \geq 0 \ \right\}$ may also be thought of as the occupancy measure of the vector $\vec{c}^N$ with values in $\mathcal{C}$. In other words, it characterizes the values taken by all the coordinates of $\vec{c}^N$ but ignores to which coordinates each value corresponds.

If we assume that the process above satisfies the initial conditions that converge to a deterministic limit $(d_c)_{c\in\mathcal{C}}$:

$$\forall c \in \mathcal{C}\,,\ \lim_{N\to\infty} u_c^N(0) = d_c\,,\ \left(\text{ for } \ d_c \geq 0, \sum_{c\in\mathcal{C}} d_c = 1 \right) \tag{3.1}$$

then as N becomes large, the Kurtz's theorem (see *e.g.,* [80]) states that the process of $\left\{ \ (u_c^N(t))_{c\in\mathcal{C}} \ \middle| \ t \geq 0 \ \right\}$ converges to a deterministic limit $\left\{ \ (u_c(t))_{c\in\mathcal{C}} \ \middle| \ t \geq 0 \ \right\}$ which is the unique solution of the following ODE problem:

$$\begin{cases} \forall c \in \mathcal{C}\,, & \frac{\partial u_c}{\partial t} = \sum_{c'\neq c} \rho_{c',c} u_{c'} - \left(\sum_{c'\neq c} \rho_{c,c'} \right) u_c \\ \forall c \in \mathcal{C}\,, & u_c(0) = d_c\,. \end{cases} \tag{3.2}$$

By Cauchy-Lipschitz theorem, for any boundary condition $(d_c)_{c\in\mathcal{C}}$ this ODE problem admits a unique solution. Following classical notation, we denote the value at time t of the solution for boundary condition d by $u_c(t|d)$.

Assuming that the matrix ρ is irreducible, we may consider the stable mobility regime where $u_c(t) = \tilde{u}_c$ independently of t and is defined as the unique solution of

$$\forall c \in \mathcal{C}\,,\ \tilde{u}_c(\sum_{c'\neq c} \rho_{c,c'}) = \sum_{c'\neq c} \rho_{c',c}\tilde{u}_{c'} \ \text{ and } \ \sum_{c\in\mathcal{C}} \tilde{u}_c = 1\,. \tag{3.3}$$

Propagation of Information

Let $A_{n,c}$, $n \in N, c \in \mathcal{C}$ be $N \times C$ independent Poisson processes such that $A_{n,c}$ has a rate μ_c. Each point of this process denotes possible information update received by user n directly from the source in class c (the transition always exist, but it has no effect unless the user n is currently in class c).

Let $B_{m,n,c}$ $m \in N, n \in N, m < n, c \in \mathcal{C}$ be $\frac{N\times(N-1)}{2} \times C$ independent Poisson processes such that $B_{m,n,c}$ has a rate $\frac{2\cdot\eta_c}{N-1}$. Each point of this process denotes a possible opportunistic contacts for the pairs $\{ \ m, n \ \}$, occurring in the class c (the transition always exist, but it has no effect unless the users n and m are currently both in class c).

Similarly, define $C_{m,n,\{c,c'\}}$ for $m \in N, n \in N, m < n, c \in \mathcal{C}, c' \in \mathcal{C}, c < c'$ be $\frac{N\times(N-1)}{2} \times$

$\frac{C\times(C-1)}{2}$ independent Poisson processes such that $C_{m,n,c,c'}$ has a rate $\frac{2\cdot\beta_{\{c,c'\}}}{N-1}$. Each point of this process denotes a possible opportunistic contacts for the pairs $\{\ m,n\ \}$, occurring when m or n is in class c and m or n is in class c' (the transition always exist, but it has no impact unless the users n and m are currently one in classes c, the other in class c').

$$\begin{aligned}
&d\vec{X}^N = \vec{1}dt - \textstyle\sum_{n\in N}\sum_{c\in\mathcal{C}} X_n^N \cdot 1_{\{c_n^N=c\}}\cdot\vec{e}_n dA_{n,c}\\
&+\sum_{m<n}\sum_{c\in\mathcal{C}}\left[1_{\{X_m^N<X_n^N\}}\vec{e}_n\left(X_m^N - X_n^N\right)\right.\\
&\left.+\ 1_{\{X_m^N>X_n^N\}}\vec{e}_m\left(X_n^N - X_m^N\right)\right]1_{\{c_n^N=c\}}1_{\{c_m^N=c\}}\cdot dB_{m,n,c}\\
&+\sum_{m<n}\sum_{c<c'}\left[1_{\{X_m^N<X_n^N\}}\vec{e}_n\left(X_m^N - X_n^N\right)\right.\\
&\left.+\ 1_{\{X_m^N>X_n^N\}}\vec{e}_m\left(X_n^N - X_m^N\right)\right]1_{\{\{\ c_n^N,c_m^N\ \}=\{\ c,c'\ \}\}}\cdot dC_{m,n,c,c'}\,.
\end{aligned}$$

We define the occupancy measure of $\vec{X}^N(t)$ in class c by:

$$M_c^N(t) = \frac{1}{N}\sum_{n=1}^{N} 1_{\{c_n^N(t)=c\}}\delta_{X_n^N(t)}\,.$$

$F_c^N(z,t)$ (*i.e.,* the fraction of users that are in class c and with ages lower than z) is

$$F_c^N(z,t) = M_c^N(t)\left([0;z]\right) = \int_0^z M_c^N(t)(du)\,.$$

3.3 Mean-Field Regime

3.3.1 Mean-Field Limit

In this section we show that as N gets large, the age evolution of the information available to mobile users becomes close to a deterministic limit characterized by differential equations. Here, we derive this result heuristically. In [16], Chaintreau et al. mathematically prove the result.

The assumption is that the initial conditions of the system, as N gets large, converge to a deterministic limit. In other words, the occupancy of classes by users converges to a deterministic vector $(d_c)_{c\in\mathcal{C}}$ according to Eq.(3.1), and the initial occupancy measure $M_c^N(0)$ of ages in each class converges weakly to a deterministic distribution m_c^0, with CDF F_c^0.

Under these assumptions, as N gets large, the collection of occupancy measures M_c^N converges in distribution to deterministic processes $\{\ m_c(t)\mid\ t\geq 0\ \}$. Furthermore, if m_c^0 admits

a density, then $m_c(t)$ has a density for all t and its CDF $F_c(z,t)$ is the unique solution of the following PDE problem

$$\begin{cases} \forall c \in \mathcal{C}\,, & \dfrac{\partial F_c(z,t)}{\partial t} + \dfrac{\partial F_c(z,t)}{\partial z} = \\ & \displaystyle\sum_{c' \neq c} \rho_{c',c} F_{c'}(z,t) - \left(\sum_{c' \neq c} \rho_{c,c'} \right) F_c(z,t) \\ & + \left(u_c(t|d) - F_c(z,t)\right) \left(2\eta_c F_c(z,t) + \mu_c\right) \\ & + \left(u_c(t|d) - F_c(z,t)\right) \displaystyle\sum_{c' \neq c} 2\beta_{\{c,c'\}} F_{c'}(z,t) \\ \forall c \in \mathcal{C}\,, & \forall t \geq 0\,,\ F_c(0,t) = 0 \\ \forall c \in \mathcal{C}\,, & \forall z \geq 0\,,\ F_c(z,0) = F_c^0(z)\,. \end{cases} \tag{3.4}$$

where $u_c(t|d)$ denotes the solution of Eq.(3.2).

This can be heuristically derived by considering the mean-field limit for the densities. The theorem implies that $F_c(z,t)$ admits a density $f_c(z,t)$ at all times t if it has one at time 0; intuitively, the density should satisfy for all c:

$$\begin{aligned} & f_c(0,t) = \mu_c \cdot u_c(t) \\ & \frac{\partial f_c(z,t)}{\partial t} = -\frac{\partial f_c(z,t)}{\partial z} - \mu_c f_c(z,t) \\ & + \sum_{c' \neq c} \rho_{c',c} f_{c'}(z,t) - \left(\sum_{c' \neq c} \rho_{c,c'} \right) f_c(z,t) \\ & + 2\eta_c \left[(+1) \times \left(u_c(t) - F_c(z,t)\right) \cdot f_c(z,t) \right. \\ & \quad \left. + (-1) \times f_c(z,t) \cdot F_c(z,t) \right] \\ & + \sum_{c' \neq c} 2\beta_{\{c,c'\}} \left[(+1) \times \left(u_c(t) - F_c(z,t)\right) \cdot f_{c'}(z,t) + \right. \\ & \quad \left. + (-1) \times f_c(z,t) \cdot F_{c'}(z,t) \right] \end{aligned}$$

The second equation can be interpreted using the different possible transitions from the point of view of the current population of users in class c and with ages around z. The first term denotes the passage of time. The second term denotes the population removed from age z by source injection. The second line denotes the movement of the user population with age z among different classes. The third and fourth line denotes the impact of opportunistic contacts within the same class and among different classes. The first transition corresponds to a new user in class c becoming of age z (which is why it is multiplied by $+1$). The user

should already be in the class c and the users it met should have age z, hence the rate of such a transition is $(u_c(t) - F_c(z,t)) \cdot f_c(z,t) \cdot 2\eta_c$, or if the contacts is among different classes $(u_c(t) - F_c(z,t)) \cdot f_{c'}(z,t) \cdot 2\beta_{\{c,c'\}}$. Last, we have to account for transition where one user in class c is not any more with age z (which explains the -1 for the population) because its age decreases. This user leaving should be of age z, and it should meet a user with an age at most z, hence the transition occurs with rate $f_c(z,t) \cdot F_c(z,t) \cdot 2\eta_c$, or respectively with rate $f_c(z,t) \cdot F_{c'}(z,t) \cdot 2\beta_{\{c,c'\}}$ if this is an opportunistic contacts with another class c'.

The above system of equations may be simplified if we write by convention, when $c = c'$, $\beta_{\{c,c'\}} = \eta_c$. We can then write, as an example, $\sum_{c \in \mathcal{C}} \beta_{\{c,c'\}} = \sum_{c \neq c'} \beta_{\{c,c'\}} + \eta_c$.

$$
\begin{aligned}
& f_c(0,t) = \mu_c \cdot u_c(t) \\
& \frac{\partial f_c(z,t)}{\partial t} = -\frac{\partial f_c(z,t)}{\partial z} - \mu_c f_c(z,t) \\
& \quad + \quad \sum_{c' \neq c} \rho_{c',c} f_{c'}(z,t) - \left(\sum_{c' \neq c} \rho_{c,c'} \right) f_c(z,t) \\
& \quad + \quad \sum_{c'} 2\beta_{\{c,c'\}} \left[(u_c(t) - F_c(z,t)) \cdot f_{c'}(z,t) - f_c(z,t) \cdot F_{c'}(z,t) \right]
\end{aligned}
$$

Note that $z \mapsto (u_c(t) - F_c(z,t)) \cdot F_{c'}(z,t)$ is a primitive with regard to z of the terms in the last sum. Therefore, after integrating with regard to z, we obtain Eq.(3.4).

3.3.2 Solution of the PDE Problem

In [16] Chaintreau et al. prove that the PDE problem described by Eq.(3.4) admits a unique solution, obtained as the transform of a function defined by an ODE problem. The unique solution F is given by:

$$
\forall c \in \mathcal{C}\ ,\ F_c(z,t) = \begin{cases} h_c(z|0, u(t-z|d)) & \text{for } z \leq t \\ h_c(t|F_c^0(z-t), d) & \text{for } z > t \end{cases} \tag{3.5}
$$

where $h(.|b,d)$ denotes the solution of the following ODE problem defined for a function

$H : [0; \infty[\rightarrow [0; 1]^C$:

$$\begin{cases} \forall c \in \mathcal{C}\,, & \dfrac{d\, H_c(x)}{dx} = \displaystyle\sum_{c' \neq c} \rho_{c',c} H_{c'}(x) - (\sum_{c' \neq c} \rho_{c,c'}) H_c(x) \\ & + (u_c(x|d) - H_c(x))\,(\mu_c + 2\eta_c H_c(x)) \\ & + (u_c(x|d) - H_c(x)) \left(\displaystyle\sum_{c' \neq c} 2\beta_{\{c,c'\}} H_{c'}(x) \right) \\ \forall c \in \mathcal{C}\,, & H_c(0) = b_c \end{cases} \tag{3.6}$$

In the special case where the class occupancy starts in steady state (*i.e.,* $u(0) = \tilde{u}$), we have $F(z,t) = h(z|0,\tilde{u})$ for $z \leq t$ and thus $F(z,t)$ does not depend on t for $z \leq t$; however, it still depends on t for $z > t$.

3.4 Validation With Traces

3.4.1 Validation Setup

We validate the model and the mean field limit using a dataset collected by Yellow Cab taxis in the San Francisco Bay Area. Thus, the role of mobile users is assigned to the taxicabs. We divide the San Francisco Bay Area into 16 classes, as shown in Figure 3.1. Fifteen classes are obtained using a regular square grid. Each of them corresponds to a region of about 4 sq km. These are the classes that cover the area of interest. The 16th class surrounds the other classes and contains the area outside classes 1-15. Its existence is important, as it keeps the number of mobile users in the system rather constant.

Base stations are placed in fixed locations, and we assume that they always have fresh information update from a source server. We assume that each mobile user (*i.e.,* a taxicab) is equipped with a short-range radio, which allows for the exchange of data upon a meeting with base stations or other mobile users. As previously explained, upon a meeting with a base station, a mobile user receives fresh information. A meeting between two mobile users results in both of them having the freshest information available in any of them prior to the meeting.

Data Sets We use GPS position records, logged approximately once per minute, which have been collected as a part of the *Cabspotting* project [81]. The project aims at visualizing the aspects of everyday life in SF. About 500 Yellow cab vehicles that operate in the area are equipped with GPS receivers. Recorded data is sent to the central dispatcher and stored in the database. Each GPS record contains the cab ID, current location, as well as the time stamp.

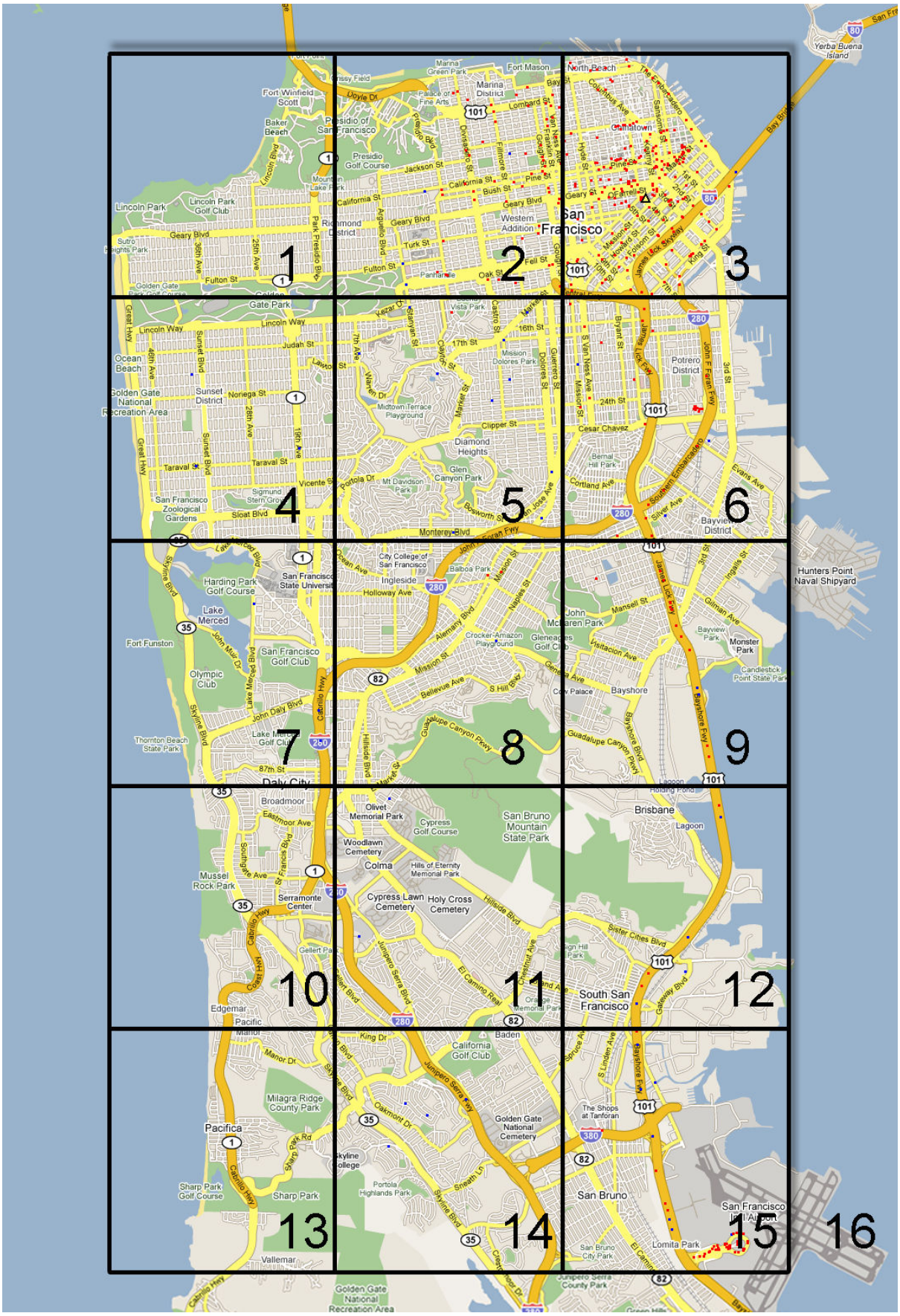


Figure 3.1: The Bay Area is split into 16 classes.

This allows us to reconstruct the path of each individual mobile node for the past two years.

We use the 30 day GPS trace, from May 17 to June 15, 2008. We observe the 16 hour periods between 8 a.m. and midnight, in order to avoid night-time, when the number of active cabs drops.

Generation of Contact Traces In order to obtain an artificial contact trace from an existing GPS trace, we first have to define ranges, both for mobile users and for base stations. We also have to define the notion of a *meeting* between two mobile users or between a mobile user and a base station. We assume that the radios of mobile users and base stations have a range of $200m$. This corresponds to the envisioned range in vehicular communications [82], and it is a bit longer than the ranges of 802.11 devices ($\sim 140m$) or Bluetooth Class 1 devices ($\sim 100m$).

Every mobile user performs scanning once per minute, looking for base stations and other mobile users in the range. Each time another mobile user or a base station is discovered, we use interpolation to make sure that the contact lasts at least 10 seconds. So, we assume that a meeting between two mobile users, or a mobile user and a base station happened if, during scanning, a mobile user detected another mobile user, or a base station and their contact lasted for at least 10 seconds. In [6], by observing an opportunistic network with buses (equipped with 802.11b radios), the authors found an average transfer opportunity duration of 10.2 seconds, which was sufficient to exchange on average 1.2MB of data.

Contacts among mobile users, and contacts between mobile users and base stations, can also occur in between the scanning periods. We decide to ignore these contacts. Given these constrains and the provided definition of a *meeting*, we run a simulation (written in Java) and obtain the contact trace.

Parameter Settings The input parameters for the model and the mean field approximation, as defined in Section 3.2, are μ_c, η_c, $\beta_{c,c'}$ and $\rho_{c,c'}$. For each class, we extract them from the contact traces as follows:

$$\mu_c(t) = \frac{N_{c,ub}(t)}{N_c(t)} \ , \ \mu_c = \frac{1}{60}\sum_{t=t_0}^{t_0+60} \mu_c(t) \, ,$$

$$\eta_c(t) = \frac{N_{c,uu}(t)}{u_c(t) * (N_c(t) - 1)} \ , \ \eta_c = \frac{1}{60}\sum_{t=t_0}^{t_0+60} \eta_c(t) \, ,$$

$$\beta_{c,c'}(t) = \frac{N_{c,c',uu}(t)}{2 * N(t) * u_c(t) * u_{c'}(t)} \ , \ \beta_{c,c'} = \frac{1}{60}\sum_{t=t_0}^{t_0+60} \beta_{c,c'}(t) \, ,$$

$$\rho_{c,c'}(t) = \frac{N_{c,c',trans}(t)}{N_c(t)} \ , \ \rho_{c,c'} = \frac{1}{60}\sum_{t=t_0}^{t_0+60} \rho_{c,c'}(t) \, .$$

where $N(t)$ is the total number of users during an observed one minute interval t; $N_c(t)$ (resp. $u_c(t)$) is the number (respectively the fraction) of users in class c during the same time interval; we denote by $N_{c,ub}(t)$ (resp. $N_{c,uu}(t)$) the number of meeting between mobile users and base stations (respectively between two mobile users) during the time interval t; finally, for any two classes $c \neq c'$, we denote by $N_{c,c',uu}(t)$ (resp. $N_{c,c',trans}(t)$) the number of meetings between users in different classes (respectively the number of transitions from c to c') during the time interval t. As shown above, per hour values of the parameters are calculated by averaging their per minute values over the period of one hour.

The values of the input parameters show that the user mobility is highly skewed: 75% of users can be found within 4 popular classes (classes 2,3,6 and 15, *i.e.,* the city center and the airport); users spend on average 12 to 40 minutes in one of these classes before moving; 10% of users can be found in the surrounding classes (*i.e.,* classes 1,4,5,9 and 12) where they spend less time (4 to 12 minutes before moving away). Class 16 contains roughly 10% of "persistent" users that remain in this class during two hours on average. All the other classes contain in total 5% of users; class 13 is normally empty. The meeting rate between any two users within the same class is typically between (1/60 minutes) and (1/80 minutes); it is higher in classes 9,12,15 (1/20 minutes), and much lower in classes 10,11,13 and 16 (bellow 1/200 minutes). Contact rates between users in different classes are often negligible (these rates are typically lower than 1/2000 minutes).

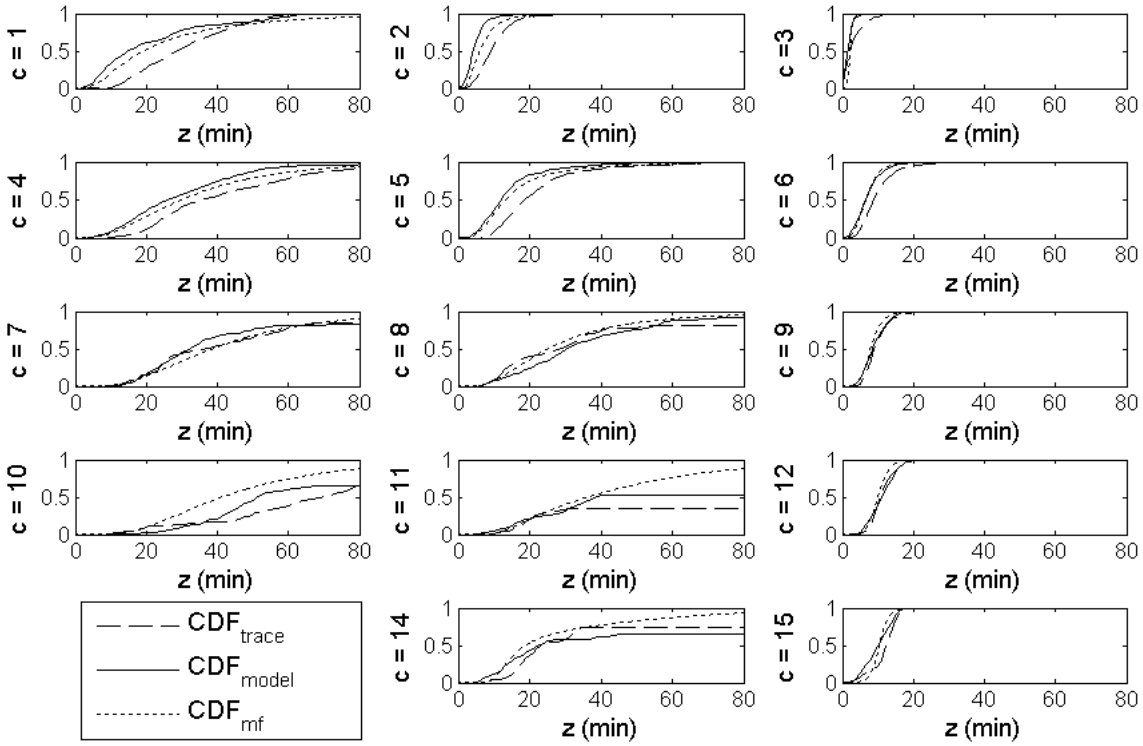
Running the Simulations The obtained input parameters are used for two purposes: (i) to simulate the random model described in Section 3.2 with $N = 500$ nodes and, (ii) to compute the mean field limit by solving the ODEs introduced in Section 3.3.2, using Matlab [1]. The contact trace itself is used directly for an event-driven simulation. In all three cases, we get the corresponding age distributions for each minute of the observation.

3.4.2 Comparison of the Trace, Model and MF Limit

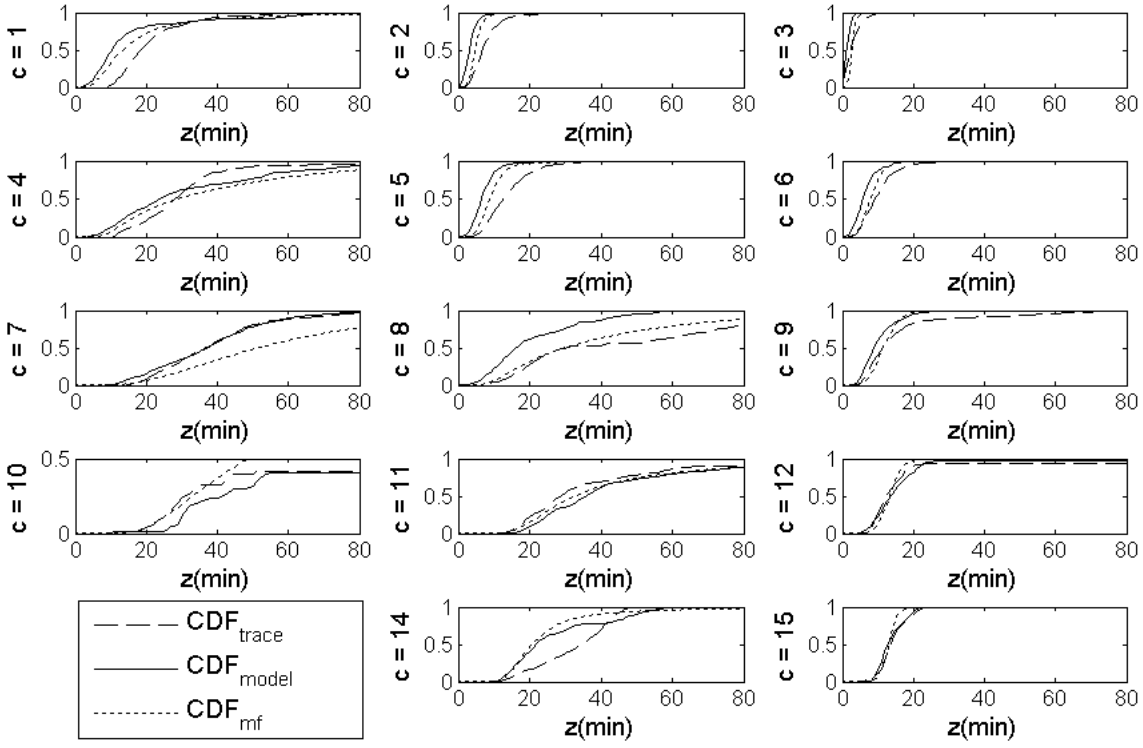
We now compare the age distributions obtained from the trace, the model and the mean-field approximation, for the case of a single base station, placed in class 3. In terms of contacts, a user in class 3 meets the base station with a rate (1/45 minutes). Simulations start at 8 a.m. We set the initial information age at each mobile user to 8 hours, in line with the night-time inactivity.

Figure 3.2 shows the Cumulative Distribution Functions (CDFs) for the ages in different

1. The value of $\tilde{u}_c$ in the ODEs is obtained from Eq.(3.3).



(a) t=1pm



(b) t=8pm

Figure 3.2: CDFs for classes 1-12 and 14-15, obtained from the trace, the model and the mean field limit. The CDFs show the age distribution (z) in different areas of San Francisco at 1 p.m. and 8 p.m., for the case of a single base station, placed in class 3.

classes, obtained at 1 p.m. and 8 p.m. from the trace, the model and the mean field limit. The hourly CDFs are obtained by averaging the CDFs for individual minutes. We omit classes 13 and 16 as they are of less interest (typically empty and disconnected from the rest of the network).

The distribution of ages, obtained from the model, shows a very good match with the distribution acquired from the trace, in particular, for the popular classes (*i.e.,* $2, 3, 6, 15$) and the classes that surround them (*i.e.,* $1, 4, 5, 6, 12$). This means that our modeling assumptions succeeded in capturing the important dynamics. Some discrepancies are observed in peripheral classes, which may be explained as follows: In classes with very few mobile users, the age of a single mobile user (which stopped for some reason too far from the main road and cannot receive an update), can create a significant difference between the trace and the model. Indeed, classes 10, 11 and 14 contain on average 1.1, 2.1 and 2 mobile users respectively.

The mean-field limit matches the model well, except, again, for discrepancies observed in peripheral areas.

The Importance of Being Opportunistic

For applications that deliver updates, the quality of service (in terms of age) can be measured as the fraction of mobile users in each class whose age is lower than a given threshold. We now compare this measure in the case where opportunistic exchanges between mobile users are allowed and in the case where dissemination is performed only via the base station.

Figure 3.3 shows the percentage of mobile users in each class that have age lower than 20 minutes at $1p.m.$ and at $8p.m.$ (300 and 720 minutes after the start of the observation), obtained from the trace, the model and the mean-field approximation. The figure also displays the same measure for the case where only the base station is used to disseminate content. We see that in this second case, the observed percentage is low and remains under 20%, even in class 3 where the base station is located. In contract, leveraging opportunistic mobile-to-mobile contacts, significantly reduces age in all classes. We observe that the fraction of users with age lower than 20 minutes in classes 2, 3, 6, 9, 12, 15 (that together contain 80% of the nodes) is very high.

These results can be better interpreted using the spatial representation shown in Figure 3.4, where data provided in the upper panel of Figure 3.3 are shown spatially for the trace and for the mean-field approximation. We observe that classes benefit differently from the base station located in class 3. Classes 2 and 6 feel the benefit as immediate neighbors. Classes 9, 12 and

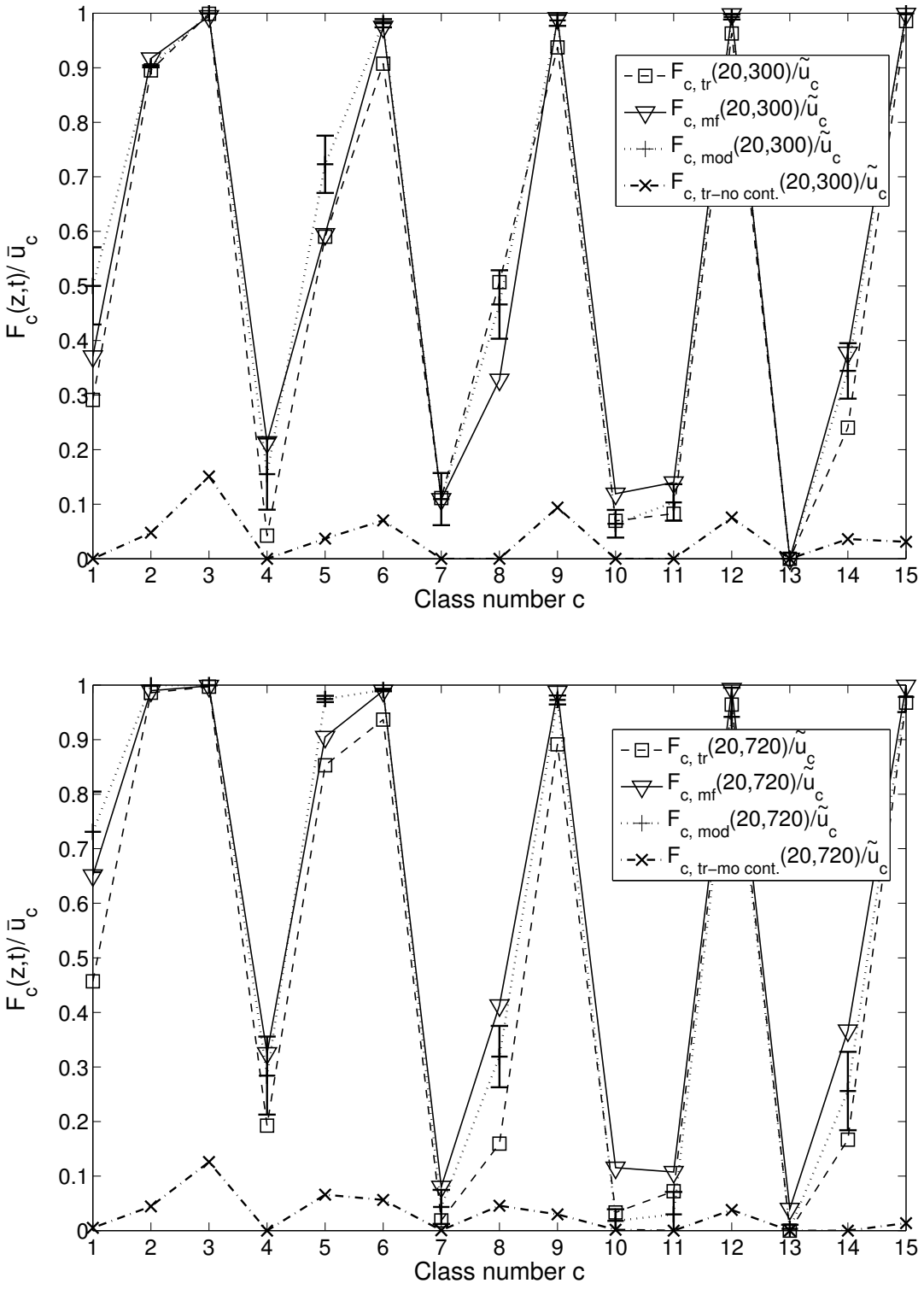


Figure 3.3: The fraction of mobile users in classes 1-15 who have age z<20 minutes, acquired from the trace, the model and the mean field limit, for a single base station placed in class 3. For the comparison, we plot the values obtained from the trace without opportunistic contacts (bottom curves). Top panel - values at 1 p.m. (t=300 minutes). Bottom panel - values at 8 p.m. (t=720 minutes).

15 benefit from the users who move along the highway between the city center and the airport, as well as from the high meeting rates among users in these classes. Classes 1 and 5, although geographically closer to class 3, benefit less from the opportunistic mobile-to-mobile contacts, due to the bias in mobility. All other classes benefit only marginally as the density of users and the contact rates in these classes are too small.

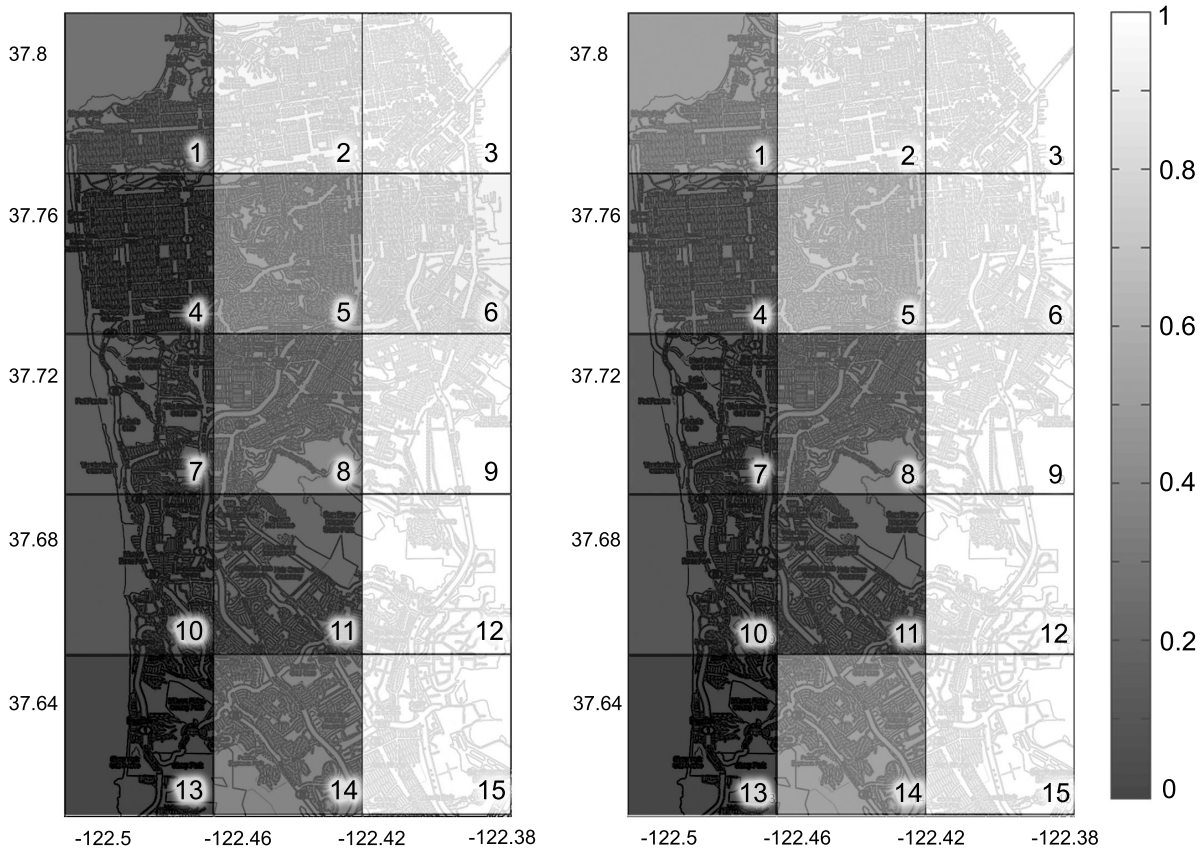
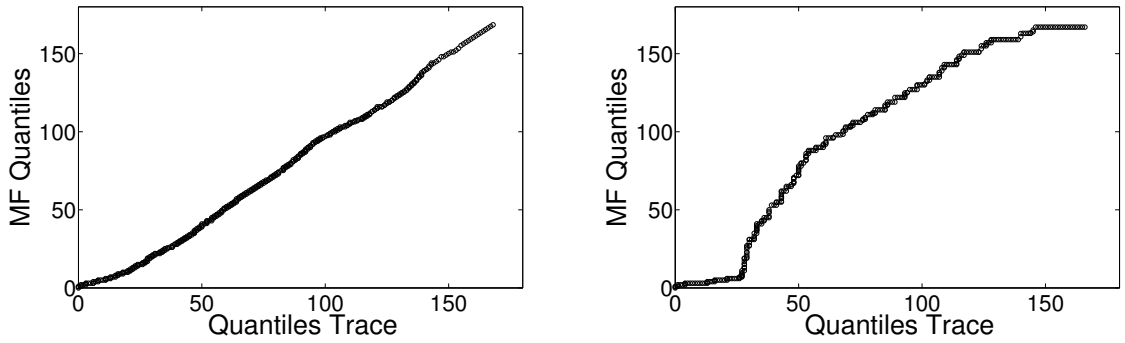


Figure 3.4: Comparison between the mean field limit and the trace: Fractions of mobile users in classes 1-15 with age z<20 minutes at time t=300 minutes (1 p.m.).

In summary, the opportunistic (mobile-to-mobile) contacts are useful as they significantly improve the availability of the up-to-date content in the network and they can compensate for a lack of infrastructure. The improvement depends critically on the user density, their mobility and the opportunistic contact rates and it is accurately captured by the mean-field limit.

The Importance of Being Spatial

We now evaluate the effects of the *spatial* approach on the accuracy of the model, by comparing our previous results (with 16 classes) with a case where classes $1-15$ have been merged



(a) MF vs Trace (C=16)

(b) MF vs Trace (C=2)

Figure 3.5: The importance of being spatial. The QQ plots, comparing the age quantiles obtained from the dataset with the quantiles obtained from the mean field CDFs, for the 16 class and 2 class scenarios. The observed time period is 5 p.m.-6 p.m.

into a single class (so that only 2 classes remain, *i.e.,* $C = 2$).

Figure 3.5 contains two QQ plots. On the first subfigure the age distribution obtained from the dataset is compared with the age distribution obtained from the mean-field approximation, for the case with $C = 16$ classes. On the second subfigure the dataset age distribution is compared with the age distribution obtained from the mean-field approximation, for the $C = 2$ class case. The dataset age quantiles for the $C = 16$ class case, are obtained from the mobile users in classes $1 - 15$, during the afternoon peak hour (5pm-6pm). The dataset age quantiles for the 2-class case are obtained from mobile users in class 1, during the same period. The mean-field ages are generated using the mean-field CDFs for the same time interval.

Figure 3.5(a) suggests that in the case with $C = 16$ classes, the mean-field limit age and the age obtained form the dataset come from the same distribution. In contrast, in the $C = 2$ class case, (Figure 3.5(b)), we observe that the mean-field limit underestimates the quantiles with low age and almost always overestimates the quantiles for high age. This is a clear indication that data comes from different distributions.

The results above show that it is essential to capture the diversity of locations (via classes), as they differ radically in terms of expected performance (age distribution). The primary factors are the dependencies between classes created by patterns of mobility (transition matrix $\rho_{c,c'}$) and the contact rates (μ_c, η_c, $\beta_{c,c'}$) that are influenced by mobile user densities and variations in placement of base stations.

3.5 Application

We now consider the following problem. We would like to leverage mobility and opportunistic contacts between taxicabs to disseminate news, traffic information or advertising. Each of these applications, however, requires a certain level of infrastructure (*i.e.,* the base stations). The number and placement of base stations, needed to achieve a certain quality of service, are not easy to guess. The answer, in general, depends on the density of users in different areas, as well as the transition rates and rates of opportunistic contacts. We show in this section that a greedy algorithm based on the mean-field limit offers a fast and efficient method for placement of base stations, over multiple classes. It also offers a significant improvement over other simple heuristics.

3.5.1 Method for Infrastructure Deployment Based on MF Approximation

The problem we try to solve can be formulated in the following way: For a fixed budget (fixed number of base stations), we would like to find an efficient placement of the base stations over a predetermined finite set of classes, based on a range of possible metrics.

Assumptions We assume that a predefined set of possible locations in each class, where the base stations can be placed, is known to the service planners. Each of these locations carries information about the popularity of the spot. Our assumption is that this piece of information, along with the other input parameters required by the model, can be provided by traffic engineers (traffic counting and estimation models), or based on a dataset, collected by some other service in the city (a taxi company for instance).

Metrics We wish to maximize one of the following objectives ($F_c(z_0, t_0)$ and $\tilde{u}_c$ follow previous definitions):

$metric_1$	$metric_2$	$metric_3$
$\sum_{c=1}^{C'} F_c(z_0, t_0)$	$\sum_{c=1}^{C'} \frac{F_c(z_0,t_0)}{\tilde{u}_c}$	$\min_{c=1,\ldots,C'} \frac{F_c(z_0,t_0)}{\tilde{u}_c}$

Maximizing $metric1$ is a global "per mobile user" objective; it tends to maximize the number of mobile users, in all classes, that have an age lower then z_0 during the peak hour (t_0); $Metric2$ is a "per class" metric; using this metric we try to achieve more even distribution of mobile users with ages lower than z_0, over the observed 15 classes. Finally, $metric3$ focuses on the class with the "worst" value of the age, and tends to decrease the gap in quality that

exists between this class and the other classes; this metric can be used for instance if we want to achieve some minimal QoS in all classes. We denote the total number of classes, where we plan to place base stations, by C'. In our particular case $C' = 15$, as we do not place any base stations in class 16 (our goal is not to improve the quality in class 16).

Class num	Number of base stations per class				
	$metr1$	$metr2$	$metr3$	$unif.$	$prop.$
$c = 1$	3	2	0	2	0
$c = 2$	0	0	0	2	6
$c = 3$	0	0	0	2	10
$c = 4$	7	5	0	2	0
$c = 5$	0	0	0	2	2
$c = 6$	1	1	0	2	5
$c = 7$	2	2	1	2	0
$c = 8$	3	3	1	2	0
$c = 9$	0	0	1	2	1
$c = 10$	2	3	5	2	0
$c = 11$	1	1	9	2	0
$c = 12$	0	0	0	2	1
$c = 13$	5	7	7	2	0
$c = 14$	6	5	5	2	0
$c = 15$	0	1	1	2	5

Table 3.1: Placement of 30 base stations in classes $1 - 15$, acquired from the *greedy placement*, which uses the ODEs for quality estimation. The results for the 3 considered metrics, as well as for uniform and proportional placements of base stations are shown.

Placement of base stations The algorithm we propose for the placement of base stations is a greedy algorithm (see [83]). Let us denote the total number of base stations by S, and the the number of base stations placed in class c by a_c. We define the cost as the total number of base station (*i.e.,* $cost = \sum_{c=1}^{C'} a_c$). As previously explained, we assume that the dependency $\mu_c(a_c)$ is known to the service planners, along with the other input parameters for our model. As defined in Section 3.2, μ_c denotes a contact rate with the base stations inside class c. We assume that a base station placed in class c cannot be seen from other classes, but only from within a region, limited by the base station's range, inside the class c. We start adding base stations one by one. For each base station there are C' possible placement options, one in each of the classes to which base stations are being added. Here we apply the *greedy* approach and use the ODEs (stemming from the mean field approximation) to evaluate which placement, out of C' possibilities, brings the most benefit to the observed metric. The base station is then placed accordingly. The procedure is repeated S times until all S base stations are placed. Algorithm requires the system of ODEs to be solved SxC' times.

Example Here we provide a numerical example for the placement method described above, based on the taxicab scenario described in Section 3.4. We assume that 30 base stations are to be placed. The input parameters for the system of ODEs are known, as well as the dependency $\mu_c(a_c)$. We use the input parameters obtained for the afternoon peak hour ($5p.m. - 6p.m.$). The goal is to choose the values of a_c for each of the fifteen classes where base stations can be placed.

Table 3.1 shows results obtained with 3 different metrics defined in this section, as well as the values of a_c for proportional and uniform placements. In the case of each metric we use $z_0 = 20\ minutes$ as the value for the target age.

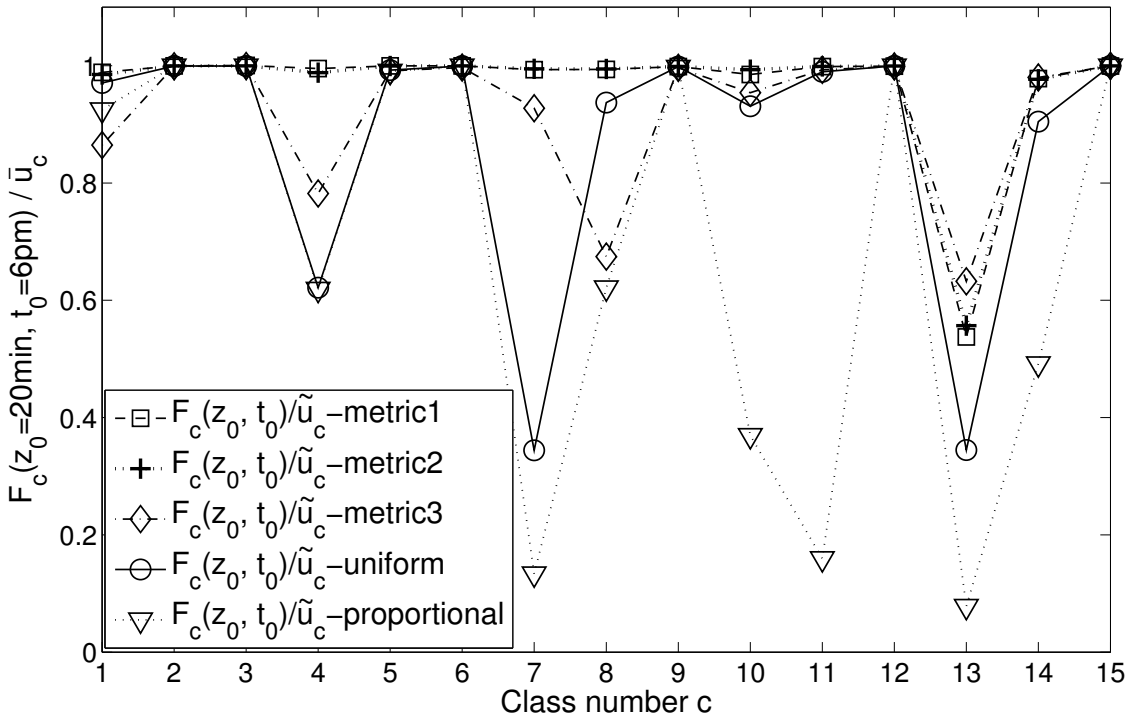


Figure 3.6: QoS achieved in classes $1-15$ using *greedy method* for placement of base stations, based on MF approximation. The curves show fractions of mobile nodes in each class with age $z < 20$ minutes, at time $t_0 = 6p.m.$, for 3 metrics, defined earlier in this section, as well as for proportional and uniform placements of base stations.

Figure 3.6 shows the effect the placements have on performance. It displays the fraction of mobile nodes in each class that have age lower then 20 minutes; we see that $metric1$ and $metric2$ provide similar results, even though the placements of base stations for these two metrics are different. $Metric3$ sacrifices efficiency for fairness. It degrades performance of

most classes, to reach a marginal improvement in the worst case class. Finally we see that proportional placement of base stations, based on the density of mobile users, results in worse performance, than the less sophisticated uniform placement with 2 base stations in each class.

3.6 Conclusion

Unlike some earlier studies that focus on latency of a single piece of content, we design a model that is capable of capturing the age distribution over all users and over different spatial regions of the network. This choice of metric allows us to observe the performance in different parts of an opportunistic network and to understand the effects of various factors that affect this performance, *i.e.,* opportunistic contacts among users, contacts with infrastructure and user mobility.

We show that, from the perspective of age distribution, areas with large number of users enter the mean field regime and we provide differential equations that describe this behavior. Finally, we demonstrate how the mean field approximation can be used as a fast simulation tool for infrastructure placement, providing the optimal placement for predefined utility functions.

Chapter 4

Opportunistic Energy-Efficient Offloading of 3G Networks

In Chapter 3, we use the system of ODEs stemming from the mean field approximation to design a method for limited infrastructure placement that boosts the performance of an opportunistic network. In this chapter, we seek to deploy wireless infrastructure and opportunistic mobile-to-mobile exchanges, in a way that would also take into account the existing cellular infrastructure. Our principal goal is to utilize the additional capacity available in opportunistic networks (based on 802.11 technology), in order to alleviate the problem of ever increasing mobile data consumption, which is putting a huge pressure on mobile operators' 3G networks. For example, in three years, the mobile data traffic in AT&T's network rose 5000% [84].

The primary reason for this situation is the rapid proliferation of smartphones, which is pushing the existing 3G networks to the limit. Although the backbone capacities are usually sufficient, it is becoming difficult and expensive for mobile operators, with a strong smartphone offering, to provide sufficient access capacity to their subscribers. After a series of reported problems [85], AT&T (until recently the only iPhone vendor in the US) purchased a $2 billion mobile bandwidth from Qualcomm in December 2010 [86].

In addition to the growth in the number of smartphones, the increase in the amount of video clips, music files and photos available on the Internet is changing the way mobile users search and access content. In two weeks, YouTube users upload 120 years' worth of movies in IMDb [87]. This user generated content is often served to users through social networks, social bookmarking services and websites for organization of social news, such as del.icio.us, Citeulike, StumbleUpon, Digg or Reddit [88].

Several studies have shown the Zipf popularity distribution of contents recommended through social networks [89]. This means that popular contents are downloaded, without constraints, by a large number of subscribers. Such behavior leads to bottlenecks, especially in densely populated urban areas, during peak usage hours. This is a strong incentive for operators to offload a part of the traffic from their 3G access networks (while preferably maintaining the ability to charge for data).

From a user's perspective, the availability of affordable data plans and the growing popularity of social networks can be mapped into a systematic overuse of battery-intensive 3G connections and an avalanche of community recommended content. Socially recommended content may not necessarily be needed in real time, however it is always treated as such and downloaded immediately via 3G at a high energy cost. For this reason, in the case of bulky socially recommended content, we propose to users to trade some delay for energy, and extend the constrained battery life of their smartphones.

We propose two algorithms for energy efficient offloading of 3G networks based on 802.11 protocol. In both cases the focus is on socially recommended, delay-tolerant content. The first algorithm, which we call the MixZones algorithm, exploits opportunistic exchanges between smartphones, in the areas called MixZones. The second algorithm, which we refer to as the HotZones algorithm, requires covering a fraction of cells, dubbed HotZones, with Wi-Fi access points. Both solutions replace a part of the costly 3G transfers with Wi-Fi transfers. In both algorithms the problem of high Wi-Fi scanning overhead is solved by the use of prediction, provided by the operator.

We evaluate the algorithms using a large, operator provided data set, which contains three months of activity and mobility for more than half a million users, in a European capital and its major commuting suburbs. We compare their performances with the real-time offloading solution, currently deployed by certain mobile operators, which allows users to seamlessly switch between Wi-Fi and 3G (we call it RT Wi-Fi Zones). For the evaluation purposes, we design a realistic application similar to Apple's Ping music social network. It allows users to request music, by relying on social recommendation, from a catalogue characterized by Zipf popularity distribution.

Our first contribution in this chapter is the design of two algorithms for delay-tolerant offloading of large, socially recommended contents from 3G networks. The algorithms take advantage of the findings made in Chapter 3 that we again confirm by using a much larger data set in this chapter. For example, although essentially different (as one algorithm relies on M2M transfers, and the other leverages Wi-Fi access points), both algorithms are based on the fact

that user mobility is biased towards certain regions, whose statistics play a particular role in the perceived performance.

The second contribution is the evaluation of the algorithms using a large operator-provided data set (with more than half a million users) that allows for a comparison of the proposed algorithms with the real-time offloading solution currently deployed by some operators. We find that both our algorithms succeed in offloading a significant amount of traffic, with a positive effect on user battery lifetime. More specifically, we show that prediction and delay (in the order of a few hours) can reduce the battery consumption coming from 3G transfers of delay-tolerant content for up to 50%. We also show that the Wi-Fi coverage needed to offload a significant amount of traffic ($80 - 90\%$) is reduced very quickly (by a factor of 3 to 4) when some delay is tolerated. Finally, we show that both algorithms deliver content with the lowest delay during the peak hours, when offloading is most needed.

Surprisingly, we find that all the benefits achieved, with the comprehensive operator-supported algorithm that relies on direct M2M transfers (MixZones), can be achieved with the less complex HotZones algorithm and a small investment in Wi-Fi access points.

The rest of this chapter is organized as follows. In Section 4.1 we present the problem background and the related work. In Section 4.2 we introduce our offloading solutions. In Section 4.3 we describe our evaluation setup. In Section 4.4 we present the performance evaluation results and in Section 4.5 we conclude the chapter.

4.1 Problem Background and Related Work

4.1.1 Mobile Data Explosion

When mobile data was introduced in the early 2000s, operators unsuccessfully looked for applications that would instigate subscribers to use slow 2.5G networks on their voice-centric phones. It was the e-mail application on the first data-centric smartphones that started to reverse the situation. The appearance of iPhone in 2007 finally changed everything and exposed users to rich data services, such as mobile video.

This event transformed the perception of mobile Internet, but it also transformed the problem of unused capacity in cellular data networks into a problem of enormous growth of mobile data traffic. Figure 4.1 compares the growth in voice and data traffic in the North America, from January 2007 to May 2009. The impact iPhone releases have on the shape of the data curve is clearly visible.

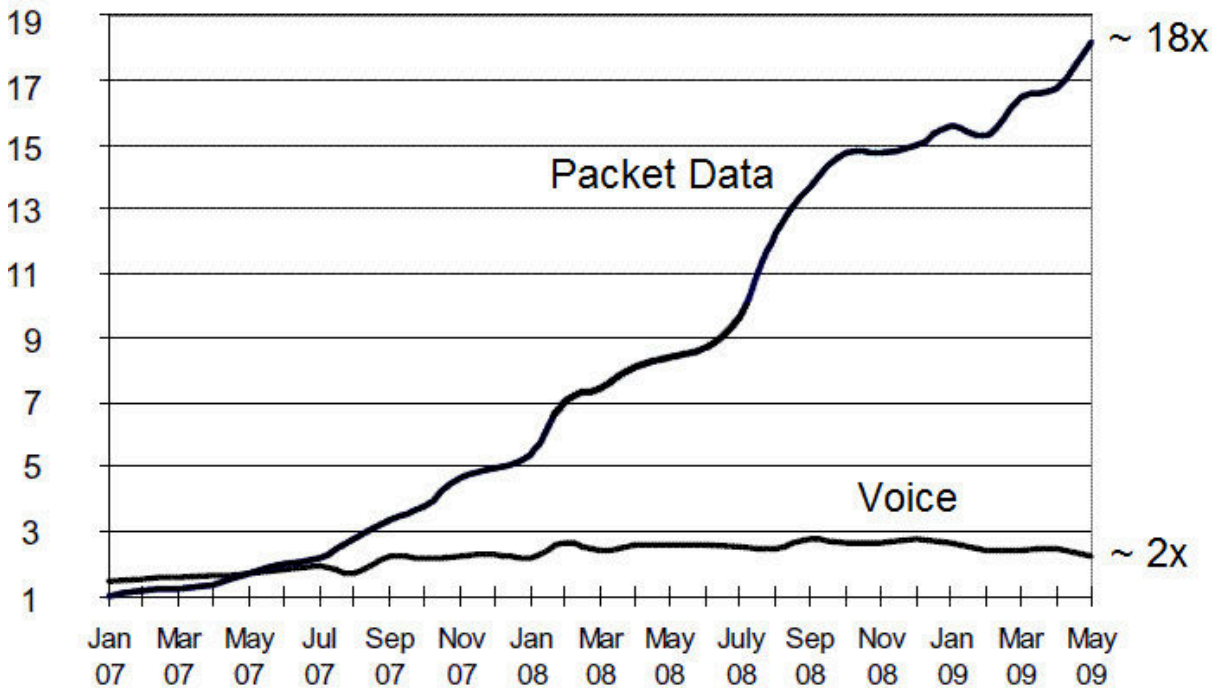


Figure 4.1: The growth of data relative to voice traffic in North America. The two inflection points for data correspond to releases of iPhone (July 07) and iPhone 3G (July 08).(Source: Rysavy Research)

4.1.2 Offloading vs. Capacity Increase

The problem the growth of mobile data traffic creates is particularly difficult to solve in the radio access part of the network. The part of the spectrum that an operator has at its disposal is limited and the efficiency of its exploitation depends on the deployed technology.

Building new cell sites or upgrading to new technologies are expensive fixes that have been applied for the past decade. The US mobile operators alone invest $50 billion in their data networks every year and the technology upgrades and innovation still fail to keep up with the demand [90]. Innovation on the other hand evolves the efficiency of transmission and reception, but it can not eliminate the fact that the number of bits that can be sent in a radio stream is limited. In spite of the continuous effort to deliver higher bandwidth over more spectral efficiency, even the new generation 4G/LTE networks are not capable of serving growing demand in the densely populated urban areas.

An alternative to capacity build-up is traffic offloading through orthogonal solutions. Licensed spectrum femtocells or unlicensed Wi-Fi can allow operators to increase capacity in certain areas, while preventing users from bypassing their networks (connections via alternative

Wi-Fi networks). Operators want to avoid bypassing and maintain control over data exchanged through the unlicensed spectrum, in order to monetize it. Since May 2010, AT&T has been deploying Wi-Fi access points in areas with consistently high 3G traffic and mobile data use [91]. We compare this solution with our delay-tolerant, prediction based algorithms.

4.1.3 The Challenges of Wi-Fi Offloading

Given the classification in Section 4.1.2, our approach can be classified as orthogonal offloading through the unlicensed spectrum, namely Wi-Fi. As previously explained, our principal goals are (i) to offload a part of the data from 3G networks and (ii) to offer users a possibility to trade delay for extended battery lifetime. We want to achieve these goals by replacing the energy costly 3G transfers with the more efficient Wi-Fi exchanges. The challenges are however the Wi-Fi's limited range, inefficient idle state and high scanning overhead.

The energy consumption of different networking interfaces present on today's smartphones depends on multiple factors, such as distance, interference, signal strength or device model. Thus, the dependency between the size of the transferred data and the energy consumed by the used networking interface is commonly obtained by averaging series of measurements at different locations, at different times of the day and by different devices [46, 47, 48, 50]. In Table 4.1 we summarize the results presented in [46] and [51].

	Transfer (J/MB)	Idle (W)	Scan (W)
3G	100	0	0
Wi-Fi	5	0.77	1.29
Bluetooth	0.1	0.01	0.12

Table 4.1: Consumption of smartphone's network interfaces. Wi-Fi transfers consume significantly less energy than 3G transfers, but scanning and idle state consumption add to the cost.

We see that, observed purely from the aspect of energy required for data transfers (and ignoring the range), Wi-Fi is much more efficient than 3G. However, any solution that requires smartphones to keep their Wi-Fi interfaces switched on, constantly scanning for transfer opportunities, would actually consume more, and not less energy than 3G transfers. Let's see it on the example of an iPhone 4 and its $5.25Wh$ battery. When switched on, an iPhone's Wi-Fi interface interchangeably scans for $1s$ and then spends $8s$ in idle state. Given the values in Table 4.1, simple calculus gives us that in this regime the daily consumption of iPhone's Wi-Fi interface would be $19.87Wh$. This means that the battery of an iPhone that performs continuous Wi-Fi scanning empties in less than $6.5h$.

So, in the ideal case, $3G$ transfers would be replaced with energy more efficient Wi-Fi transfers whenever possible, but Wi-Fi interfaces would be switched off whenever transfer opportunities are not present. In other words, Wi-Fi craves for alternative solutions for the detection of transfer opportunities. In order to solve this problem, we use prediction provided by the operator.

4.1.4 Related Work

Comprehensive measurement studies of energy consumed by wireless smartphone interfaces were performed in [46, 47, 48]. They all show that Wi-Fi scanning and idle state have rather high power consumptions, which means that continuous Wi-Fi discovery quickly drains the phone battery. In terms of data transmissions, they show that 3G transfers consume significantly more energy than Wi-Fi transfers, which is understandable given the radio ranges of these technologies.

For these reasons, a number of papers proposed modifications in the usage strategies of different wireless interfaces that are present on mobile devices. This body of work typically exploits the diversity of smartphone interfaces and mobility, with the goal of improving energy efficiency and/or download speeds. In [49], the authors propose collaborative downloading as means of increasing download speeds and battery life. In [50], policies for switching between multiple interfaces are proposed, with the goal to increase battery lifetime. Namely, the authors propose switching between Wi-Fi and low-power Bluetooth radio during data transmissions. Ananthanarayanan et al. [51] try to improve the energy efficiency of Wi-Fi by replacing Wi-Fi scanning with Bluetooth scanning. To the best of our knowledge, we are the first to estimate the energy saving achievable by cellular subscribers, coming from the use of opportunistic bandwidth. Our study is also the first to quantify the amount of data traffic that can be offloaded from a 3G network using the opportunistic bandwidth for different values of delay (different QoS requirements).

Most studies that leverage the diversity of wireless interfaces to save energy (including ours) require occasional wake-ups of the wireless interfaces, which are often asleep for power efficiency reasons. The existing proposals typically require certain modifications of the existing infrastructure. The exception is the work by Wu et al. [52]. They use cellular footprints to wake up the Wi-Fi interfaces on smartphones when in proximity of Wi-Fi APs. In [53], Agarwal et al. propose the use of a modified VoIP gateway that would turn on Wi-Fi whenever a VoIP call arrives. Closer to our work is the proposal by Shih et al. [54], who use a separate paging

infrastructure to wake up Wi-Fi.

Another related body of work concerns studies of human mobility [92]. In [40] and [41], the authors use operator provided data to show that contrary to common beliefs, humans follow repetitive and reproducible patterns. We show how this predictability is a key to solve the issue of energy efficient 3G data offloading. In [93], the authors investigate the correlation between locations and types of users' activities (types of content that they access). This mapping between user mobility and activity can be used to further improve prediction and to simplify the identification of locations where Wi-Fi can assist 3G data transmissions.

Finally, closely related to the application analyzed in this chapter are the applications that leverage the cloud to surpass the limitations of mobile environment [94].

4.2 Our Offloading Solutions

In this section we first describe the two algorithms for delay-tolerant offloading of 3G networks that we propose, namely, the HotZones algorithm, based on the placement of Wi-Fi access points and the MixZones algorithm, based on direct operator-guided M2M transfers. We then discuss the implementation aspects of these algorithms in Section 4.2.3 and the technique used to extract user mobility from the used call detail records (CDRs) in Section 4.2.4

4.2.1 HotZones Algorithm

A HotZone is a cell, partly covered by the operator owned Wi-Fi access points. We do not expect this coverage to be perfect. Thus, when in a HotZone, a user can expect to receive a requested content through one of these access points with probability p. We assume that an operator deploys the Wi-Fi access points in addition to the existing 3G infrastructure, with the goal of offloading a part of the traffic from the 3G network.

In the process of HotZones selection, an operator first extracts typical mobility profiles of its subscribers. We refer to these profiles as User Mobility Profiles (UMPs). The process of their extraction is described in Section 4.2.4. With the UMPs created, an operator ranks cells based on the average number of daily visits. Then, a set of HotZones H is chosen in a greedy way, so that a cell with the highest number of daily visits is added first to the set, the second most visited cell is added second, etc. The cardinality of the set H is a tradeoff between the cost of the Wi-Fi deployment and the targeted benefits. As we show in Section 4.4, this number strongly affects the observed performance measures.

Procedure 1 Serving user's requests in a network with HotZones.

if $(S_r^u(t) \neq \emptyset)$ **then**
  if $(c \in H)$ **then**
    $Turn\ on\ WiFi\ interface$;
    $Try\ to\ serve\ all\ r \in S_r^u(t)\ via\ WiFi$;
    $//a\ success\ with\ probability\ p$
  else
    $Get\ \tau_H^u = time\ before\ u\ enters\ a\ cell \in H$;
    for all $r \in S_r^u(t)$ **do**
      if $(\tau_r\ expires\ in\ \leq \tau_H^u)$ **then**
        $Serve\ r\ via\ 3G$;
      else
        $Do\ nothing$;
      end if
    end for
  end if
else
  $Do\ nothing$;
end if

The rationale behind the greedy selection of HotZones is that a user's request does not have to be served in a cell where it is created. As we target delay-tolerant offloading (keeping in mind that Wi-Fi access points are affordable, but not free) it makes sense to concentrate on cells with a high number of daily visits.

Once the set of HotZones H is created, an operator sends it to each user, along with his UMP. The operator can also send occasional updates if needed (for example if a new cell is covered by Wi-Fi access points). As any mobile application can obtain the real-time information about the current cell, it can use the set of HotZones and the UMP for the prediction of Wi-Fi availability. A whole class of mobile applications, where delay-tolerant content is requested can benefit from such prediction.

One such application, which we use in our evaluation is described in Section 4.3.2. Let us denote by $S_r^u(t)$ the collection of pending requests of a user u (*i.e.,* the user u's requests that are still not served at time t). Let us denote by c the current cell of the user u. Finally, let us denote by D the maximum delay users permit. Each time a request r is created, a timer τ_r with timeout equal to D is set by the application. If the request is not served before the expiry of the timeout, it is served via 3G. Using these parameters, the application on user u's smartphone performs $Procedure\ 1$ every T_P minutes.

We see that the application relies on the user's UMP for the prediction of possible Wi-Fi transfer opportunities within the allowed delay D (enforced with timers τ_r). If such an opportunity is not likely to emerge, the pending requests in the set $S_r^u(t)$ are served immediately through $3G$ in order to minimize delivery delays.

4.2.2 MixZones Algorithm

A MixZone is a (c, t) pair (where c denotes a cell and t denotes an hour of the day). The set of MixZones M is selected by an operator using the following probabilistic geometric model. Let us denote by A_c^e the effective area of a cell c. Let us denote by R the Wi-Fi radio range ($90m$) and let us denote by $N_c(t)$ the number of users in cell c during hour t. A pair (c, t) is added to the set M if, on average, the following condition is satisfied for hour t:

$$p_c(t) = 1 - (1 - \frac{R^2\pi}{A_c^e})^{N_c(t)} \geq P_{thresh}$$

Probability $p_c(t)$ is an estimate of the probability that a user in a cell c enters the range of another user during hour t. We assume that the spatial distribution of users in cell c is uniform. P_{thresh} is the value of the probability $p_c(t)$, which needs to be exceeded at hour t in order for the pair (c, t) to be added to the set M. The effective area of the cell A_c^e is introduced to compensate for the assumption of uniformity, as there are regions in each cell that are less likely to be visited by users. Thus, A_c^e represents 90% of the cell area in the case of small cells ($A < 4km2$), 75% in the case of medium cells and 60% in the case of large cells ($A > 25km2$).

The HotZones algorithm has only the spatial dimension. With the MixZones algorithm, we also have the temporal dimension. A cell that is a MixZone at time t_1 is not necessarily a MixZone at time $t_2, t_2 \neq t_1$. This is because the MixZones algorithm is based on opportunistic transfers, which means that users that want to exchange content have to be in radio range, with their Wi-Fi interfaces switched on *during the same period of time*. As we want to avoid the Wi-Fi scanning, it is the operator who decides when and where the Wi-Fi interfaces on a group of users' devices are switched on.

In the case of the MixZones algorithm the quasi-static user mobility profiles (UMPs) are not sent to users. Instead, an operator uses UMPs, along with the set M, to concurrently signal to a group of users' smartphones if their Wi-Fi interfaces need to be switched on. As UMPs and set M are stored only on the operator side, they can be refreshed more often (than in the case of the HotZones algorithm), using the information coming from calls and data sessions.

As the MixZones algorithm is based on opportunistic exchanges, it is assumed that every

Procedure 2 Serving user's requests in a network with MixZones.

if $(c \in M)$ **then**
  $Turn\ on\ WiFi\ interfaces\ in\ set\ U_c(t)$;
  $Opportunistic\ WiFi\ transfers\ among\ users$;
else
  for all $users\ u\ in\ cell\ c$ **do**
    $Get\ \tau_M^u = time\ before\ u\ enters\ a\ cell \in M$;
    for all $r \in S_r^u(t)$ **do**
      if $(\tau_r\ expires\ in\ \leq \tau_M^u)$ **then**
        $Serve\ r\ via\ 3G$;
      else
        $Do\ nothing$;
      end if
    end for
  end for
end if

user has a cache, where she stores content that can be sent to other users. Additionally, it is assumed that an operator has the real-time insight in the content requested by users and content available in users' caches. Whenever a user creates an item request or receives an item, she notifies the operator's cloud, by sending the ID of the item.

Similarly as in the HotZones algorithm, let us denote by $S_r^u(t)$ user u's collection of pending requests. Let us denote by D the maximum permitted delay. Each time a request r is created, a timer τ_r with timeout D is set by the application. If the request is not served before the expiry of the timeout, it is served via 3G. Finally, given the knowledge of items requested by users and items available in their caches, at any time t and in any cell c, an operator can select a set of users $U_c(t)$, such that each selected user: (i) either has items requested by some other users in c or (ii) requests items available in the caches of some other users in c. Using these parameters, a server in the operator's cloud performs $Procedure\ 2$ every T_P minutes, for every cell c in the network.

The idea behind the creation of the set U_c is to switch on only the users that can contribute to data transfers. The problem is similar to the NP hard *set cover problem*, where a set of items is to be covered with a number of subsets. It differs in that in our case each requested item should be covered by preferably more than one copy, in order to increase the delivery probability.

4.2.3 Implementation Aspects

From the implementation aspect, HotZones algorithm is less complex to deploy. It requires an operator to create UMPs and the set H and to deliver them to users. Apart from this initial support from the operator (and possible occasional updates), the HotZones algorithm is completely distributed. All decisions with regard to the use of networking interfaces are made locally by the smartphone application. The APIs of today's smartphone operating systems (such as iOS) enable applications to switch between 3G and Wi-Fi. An interworking WLAN client application on the handset offers the ability for two functions. The switchover is seamless and presents a transparent view to the user.

In the case of MixZones algorithm, support for ad hoc exchanges between users' smartphones is needed. Such support exists in the case of iPhone and it is additionally improved with the release of the iOS 4.3 software update.

Regarding the operator's assistance, MixZones algorithm is more demanding. First, an operator is required to maintain a fine-grained knowledge of users' requests and caches, in order to avoid switching on Wi-Fi interfaces on devices that can not contribute to data exchanges. This task can be performed by a server in the operator's cloud. The server can receive small incremental updates, sent by users, following the changes in their caches or requests. The updates, containing only item IDs, can be uploaded via 3G. Due to their small size, they would consume few resources.

Second, MixZones algorithm requires an operator to switch on Wi-Fi interfaces on users' smartphones remotely, so that a group of users in a cell have their Wi-Fi interfaces turned on during a same time period. There are multiple possible solutions to this problem. One of them is the use of control channels. In order to quickly locate called users, base stations maintain communication with subscribers, even when they are inactive. Cell phones send location updates to base stations through the access channel and base stations occasionally page users using the paging channel. Control channels are also used for sending text messages and similarly, an operator can use them to signal to a smartphone if a networking interface needs to be switched on.

4.2.4 Inferring Users' Mobility

The most commonly stored users' activity (and mobility) records are Call Detail Records (CDRs). A CDR contains calling and called users' numbers (blank in the case of a data session), date and time, session duration, caller's cell ID, cell coordinates, etc. As explained in

Section 4.3.1, these are precisely the records we have at our disposal.

As both proposed algorithms rely on users' mobility, we use CDRs to obtain it. The approaches to describe users's mobility can be classified as: (i) quasi-static, where a rather permanent list of pre-computed locations describes the mobility of a user and (ii) dynamic, where a list of cells is dynamically adjusted (with expiry of cells).

Using only one month of the data set, we extract what we refer to as quasi-static, user mobility profiles (UMPs). A UMP is an array of 24 elements, which contains the most visited cell by a user for each hour of the day. For each of the half of a million users that we observe, we extract two such profiles, one for the weekdays and one for the days of the weekend. We use the remaining two months of data to test how accurately UMPs predict users' mobility. With only one month of data used for the creation of profiles, we obtain a 69% match with the remaining two months. This relatively high prediction accuracy, based on a few weeks of data, is the result of a high correlation between daily mobility patterns of individual users, especially for the weekdays. Users tend to visit the same cells at the same hours.

Once an operator has the UMPs extracted, these can be sent to users (*i.e.,* every user is provided with her own mobility profile). Although we find that UMPs show little change over time, it is possible for an operator to occasionally recalculate UMPs. This way, the quasi-static mobility profiles can be made more dynamic and adjustable to possible changes in mobility, which can occur over time (the change of workplace, address, etc.).

Our algorithms use UMPs for the prediction of upcoming areas suitable for Wi-Fi transfers, where switching between networking interfaces should occur. More generally, the mobility profiles can be used by a wider range of smartphone applications (for example, any application that sends push notifications to users based on expected mobility).

Finally, from the perspective of HotZones algorithm, it is interesting to check if most users generate their requests from a small subset of frequently visited cells. Unfortunately, our data shows that this is not the case. By observing only data session CDRs, over the period of three months, we can see that users tend to download content from a wide range of locations. Similarly, by focusing on MMS records we can see that uploads [1] are made from a variety of locations. Although users request content from a variety of locations, we notice that a relatively small subset of cells reoccurs in the majority of UMPs. These are precisely the highly frequented cells that are top candidates for HotZones.

CDRs are not the most detailed location logs an operator can store. Every cellular operator has access to more detailed location records. They contain information exchanged via the

1. Uploads are less important for the application that we consider.

paging and access channels. Log files containing this additional information would allow us to recreate UMPs with more accuracy. However, from the aspect of our goal, CDRs seem to naturally fit the purpose. They permit us to observe mobility through activity and as the goal is an activity driven offload, what is needed are the areas with high user activity.

4.3 Evaluation Setup

4.3.1 About the Data Set Used in the Study

The data set that we use is obtained from a major mobile operator and it consists of CDRs for 1 million users for a period of three months (October-December 2009). The data covers an area of a Western European capital and its major commuting suburbs. We focus our analysis on $533,189$ users, which had more than 50 records (calls/data sessions) per month.

4.3.2 Social Music Sharing Application

In order to estimate the proposed algorithms' potential for offloading of socially recommended contents, such as music or video, we consider an application that allows users to request media items based on social recommendation. All items belong to a catalogue of size I, characterized by Zipf(1) distribution. It was shown that Zipf distribution describes content popularity in many social and content sharing networks; a recent study of del.icio.us [95] found Zipf distribution in tags associated with the URLs flickr.com (photos), del.icio.us (social bookmarking), pandora.com (music) and youtube.com (video). The same distribution was found to describe the popularity of YouTube videos in [87].

We assume that each user has a cache (a library) with b items. The caches are refreshed following one of the three popular caching strategies: FIFO, LRU (Least Recently Used) and LFU (Least Frequently Used). The LRU and LFU algorithms are completely distributed; they are based only on a user's local observations of the requests for items in his cache.

The total of N users request items following two request dynamics: (i) every time a user A calls a user B, he requests an item from B's cache, with the constraint that the item is not already requested by user A or that it is not in his cache; (ii) every time a user initiates a data session he requests an item from the catalogue, following the Zipf distribution of items' popularities.

Given these request dynamics, at each moment in time t, the state of a user u is described by: (i) the current cell c, (ii) the collection of pending requests $S_r^u(t)$ and (iii) the collection of

available items $S_a^u(t)$ (*i.e.,* the b items in the user's cache).

The described application has certain similarities with two recent Apple projects. In May 2010 Apple filed a patent application that describes a system for targeted ads based on the contents of friends' media libraries. In September 2010, Apple added a music social network to iTunes, called Ping, that enables users to share music preferences with friends [96].

4.3.3 Trace Driven Simulation

We design a Java simulation framework that enables us to perform discrete event simulations, exploiting the real user mobility and activity, extracted from the data set described in Section 4.3.1. The framework permits, at any moment in time, to keep track of users' states, *i.e.,* their current cells, the contents of their caches ($S_a^u(t)$) and the lists of their requests ($S_r^u(t)$). The requests come as a result of real calls and data sessions initiated by users. The framework allows us to simulate different caching strategies and different cache sizes.

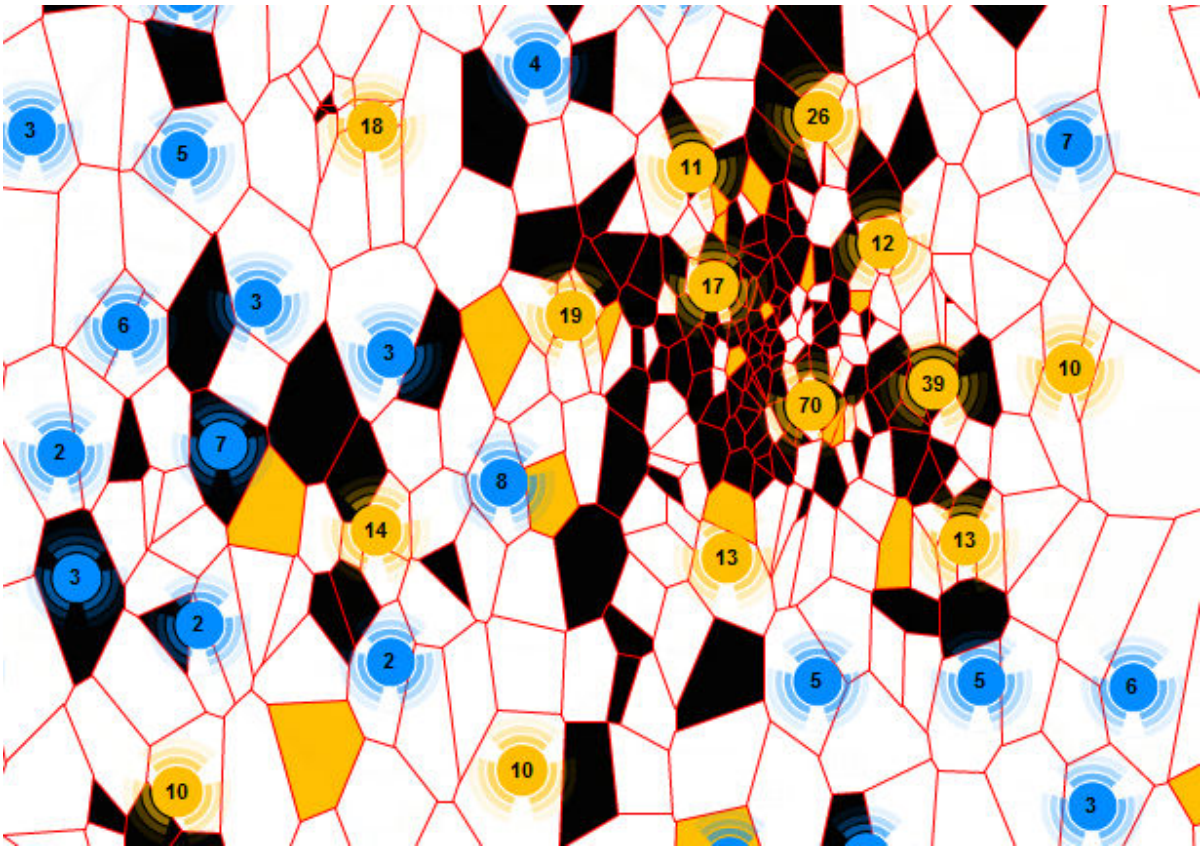


Figure 4.2: HotZones (yellow cells) and the cells that form the MixZone pairs (black cells) in the city center. The circle-shaped markers contain numbers of cell towers in different regions.

We simulate the proposed algorithms with $N = 533189$ users, who move between $C =$

1141 cells, following their real mobility recorded in the CDRs. The simulation lasts for 30 days, which are different from the month used to extract UMPs.

With both algorithms, the turning on of a Wi-Fi interface incurs the energy cost of two $1s$-scanning intervals and $8s$ of idle state, even if no data transfers occur. The energy consumption is calculated using the values in Table 4.1. The parameter Tp is set to $10min$.

There are $I = 100K$ items in the considered media catalogue. The popularity of items follows Zipf(1) distribution. Users' caches are initially filled with items following the same distribution (*i.e.,* the probability that an item is found in a user's cache depends on its popularity obtained from Zipf(1) distribution). The caches remain full throughout the simulation. The items in them are replaced following one of the caching strategies. The item size is uniformly distributed between 5 and $10MB$, which is comparable to the size of a large music file or an average YouTube video. We run simulations with cache sizes of $b = 100$ and $b = 1000$ items, which corresponds to $0.75 - 7.5GB$ of storage space. The media catalogue size can be compared with sizes of large music catalogues (such as iTunes), whereas the simulated cache sizes are a reasonable estimate of the sizes of personal smartphone media libraries.

We first infer the set of MixZones by setting the parameter P_{thresh}. The choice of this parameter is conditioned by the energy efficiency requirement. As shown in Figure 4.9, the value $P_{thresh} = 0.8$ saves most energy and allows to 225 cells to form 2612 MixZone (c, t) pairs. Next, we look for the set of HotZones that provides comparable performance to the HotZones algorithm. We find that the top 30 cells, selected following the procedure described in Section 4.2.1 meet this goal. The HotZones specific parameter p (which denotes the probability that a request is served via a Wi-Fi access point in a HotZone) is set to 0.9. The HotZones in the city center, and the cells that participate in the MixZone pairs, are shown in Figure 4.2.

We simulate both algorithms with the value of parameter D (maximum permitted delay) equal to $1h$, $6h$ and $24h$. In order to evaluate the impact of prediction and delay tolerance, we also simulate the special case of the HotZones algorithm, with $D = 0$, which we dub the Real-Time Wi-Fi Zones. This solution is currently considered (or deployed) by a number of operators. Smartphones (such as iPhone) support it with seamless switchover between 3G and Wi-Fi.

4.4 Performance Evaluation Results

In this section we show the results for the amounts of saved energy and offloaded traffic, obtained using the data set and the evaluation setup introduced in Section 4.3. We also look

for the principal factors that influence theses measures, *i.e.,* the major sources of improvement. We then compare the nominal delay in the system (set through parameter D) with the effective delay observed for different times of the day. We also compare the resources required to perform delay-tolerant offloading with the resources needed for real-time offloading. Finally, we discuss the process of MixZones selection, which balances between the amount of offloading traffic and energy efficiency.

4.4.1 Energy Saving and Offloaded Traffic

Both algorithms achieve significant energy saving. Up to 75% of traffic offloaded by only 30 HotZones. For the selected sets of HotZones and MixZone pairs, we plot the traffic offloaded from the 3G network and the amount of energy saved, as a function of the maximum permitted delay D (Figure 4.3). We see that for $D = 1h$, roughly 20% to 40% gets offloaded to Wi-Fi and 20% to 35% less energy is consumed by the application. For $D = 6h$, this fraction goes up to 50%. In the case of $D = 24h$, the impressive 60-75% are offloaded with as few as 30 HotZones.

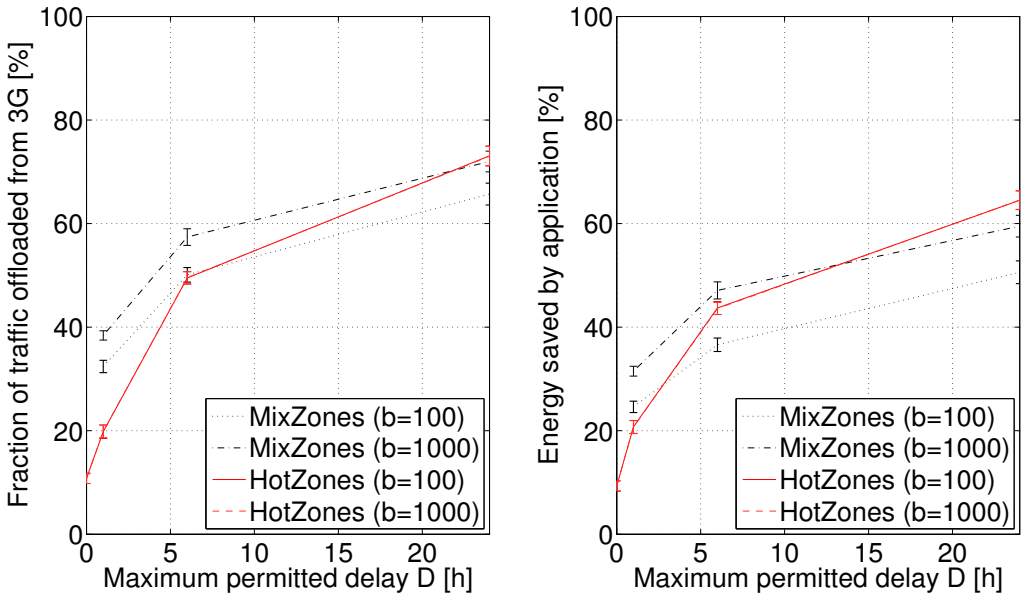


Figure 4.3: Offloaded traffic and saved energy as a function of the maximum permitted delay D. The curves are obtained for the MixZones (225 cells) and HotZones (30 cells) algorithms, with LRU caching strategy. The curves for the HotZones algorithm practically overlap.

We can also see that the HotZones algorithm is less efficient than the MixZones algorithm in the case of lower permitted delays ($D = 1h$). This is because one can not expect users to enter

one of the very few HotZones every 60 minutes. However, as the permitted delay increases, users become more likely to enter the HotZones and the performance of the algorithm improves.

4.4.2 Effects of Caching

Caching strategy has little effect on performance. Cache size is crucial for the MixZones algorithm. One of the first things we observe is that for the mobility and the request dynamics obtained from our data set, caching strategies have a very limited effect. With an average of 80 requests per month, and the user cache sizes of $b = 100$ and $b = 1000$, the initial Zipf(1) distribution of items in users' caches is well maintained after 30 days, for all three simulated caching strategies. The Complementary Cumulative Distribution Functions (CCDFs) in Figure 4.4 show the initial distribution of items in users' caches and the distributions after 30-day simulations, with LRU, LFU and FIFO caching strategies. We see that even with the caches of $b = 100$ items, the system stays stable. Consequently, the values of the performance metrics that we obtain for these caching strategies are very similar. In order to avoid the unnecessary repetition, in the rest of this section we show the results for the LRU caching strategy only.

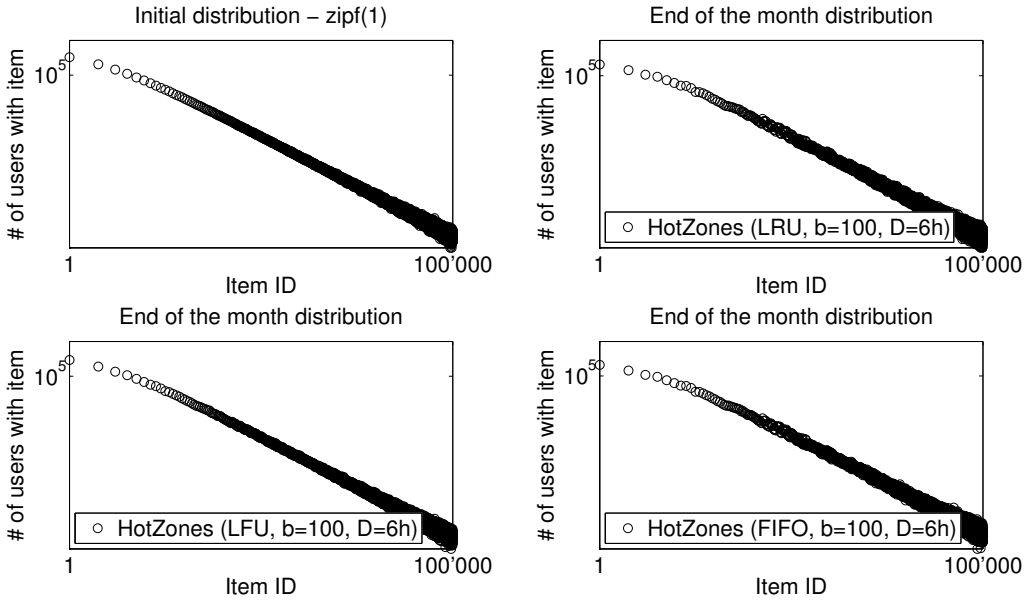


Figure 4.4: Evolution of item popularity distribution: The four curves are plotted on the log-log scale and they show the initial distribution of items in users' caches and the distributions after 30-day simulations, with LRU, LFU and FIFO caching strategies.

Unlike caching strategies, the cache size plays a major role in the case of the MixZones (Figure 4.3). Larger cache sizes increase the probability that an encountered user can serve a

request. Hence, the improvement brought by the cache size, is significant. On the contrary, as expected, the cache size does not affect the HotZones algorithm, where serving requests depends only on users' mobility and the selected set of HotZones. Thus, the curves for $b = 100$ and $b = 1000$ almost overlap.

4.4.3 Sources of Energy Saving

Most energy saving comes from prediction and delay tolerance. The special case of the HotZones algorithm with $D = 0$ (which we refer to as the RT Wi-Fi Zones), allows us to estimate the offloading and energy saving that do not come from prediction and delay tolerance, but purely from the placement (addition) of Wi-Fi access points. As we see in Figure 4.3, with 30 HotZones and $D = 0$, only about 10% of traffic is offloaded and about the same amount of energy is saved. This means that the rest of the improvement observed for higher values of D comes from prediction and delay tolerance.

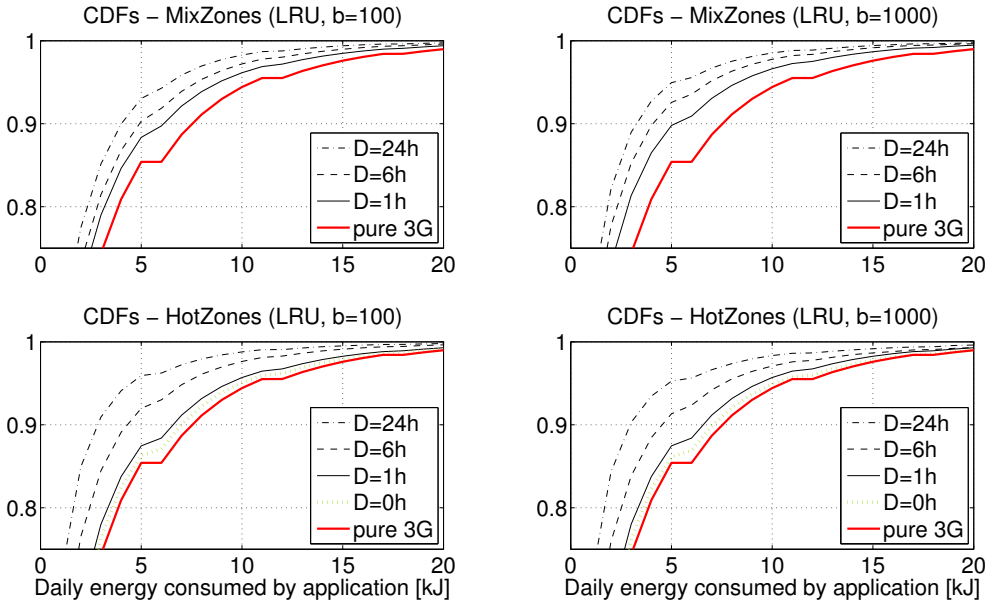


Figure 4.5: Cumulative Distribution Functions (CDFs) for the daily energy consumption of the application. The subfigures correspond to different combinations of the offloading algorithm and cache size b.

The energy improvement brought by prediction can be better observed in Figure 4.5. The figure contains the Cumulative Distribution Functions (CDFs) for the daily energy consumption of the application, for both evaluated algorithms and cache sizes. The dotted curve in two bottom figures is the CDF for the case of RT Wi-Fi Zones. We see that it almost overlaps with

the CDF coming from pure 3G transfers, yielding less than 10% improvement (as shown in Figure 4.3). Again, the increase in cache size affects only the MixZones algorithm.

In order to better understand the origin of the energy savings with MixZones and HotZones, we plot two histograms that show the energy consumed to serve users' requests (Figure 4.6). We see that in the case of both algorithms a portion of requests is served via 3G. The energy required to serve such a request ranges from 500 to $1000J$, depending on the item size (as explained in Section 4.3.3, item sizes are uniformly distributed between 5 and $10MB$). In the case of a pure 3G delivery (without either of the proposed algorithms) only this part of the distribution would exist.

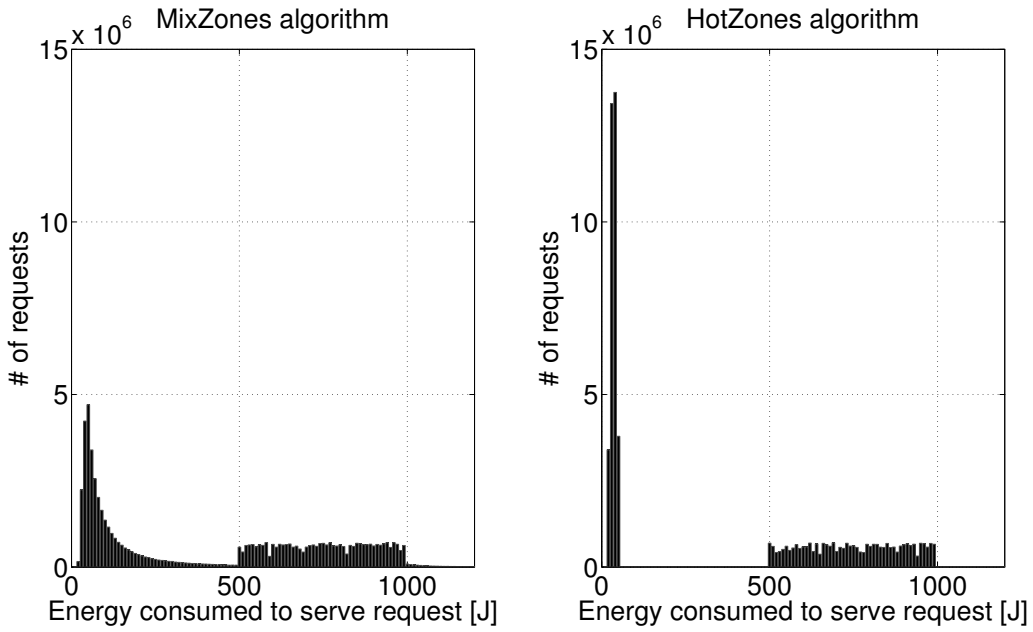


Figure 4.6: Request energy histograms show energy consumed to serve users' requests. The uniform portion on the right comes mostly from the requests served via 3G. The modes on the left come from the requests served via Wi-Fi.

However, with MixZones and HotZones algorithms, we observe a mode on the left, which comes from the requests served via Wi-Fi. In the case of HotZones, the mode is formed around the value that corresponds to the energy needed for a single item download via a Wi-Fi access point, plus the energy needed for switching on a Wi-Fi interface. In the case of the MixZones algorithm, the mode is moved towards the value corresponding to two Wi-Fi transfers (sending and receiving users), plus the energy cost of turning on of two Wi-Fi interfaces. Additionally, in the case of the MixZones algorithm, this part of the distribution is more skewed, as it is more likely that a user, with her Wi-Fi interface turned on, would miss a transfer in a MixZone, than

in a HotZone. This comes from the fact that a user in a HotZone finds an access point (with access to all items) with probability $p = 0.9$ and a user in a MixZone meets another user (with only b items) with probability $P_{thresh} = 0.8$. Thus, the MixZones algorithm sometimes requires users to have their Wi-Fi interfaces switched on several times before a request is served.

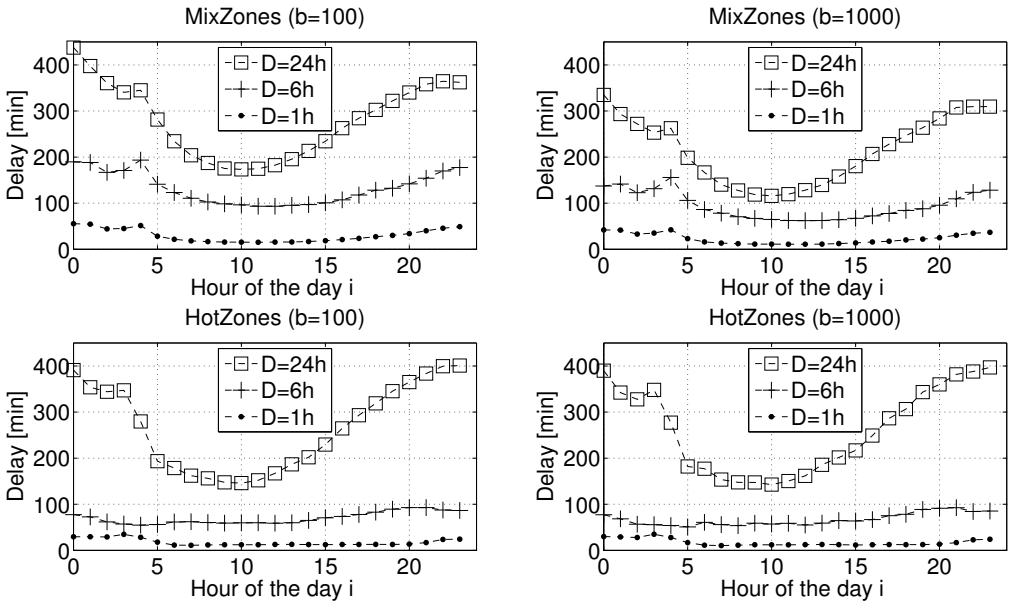


Figure 4.7: The average delay with which requests are served as a function of the time of the day and the maximum permitted delay D.

4.4.4 Effective Delay in the System

Effective delay in the system is much lower than the maximum permitted delay D. Another important performance metric is the delay with which users' requests are served. The maximum permitted delay D sets the upper limit on item delivery time. However, as we can see in Figure 4.7, the average time with which users' requests are served is often much lower than the value of D. For $D = 24h$, the requests are actually served in less then $7h$, and as fast as $2h$ during some periods of the day (depending also on the algorithm used). In case of $D = 6h$, the actual delay is between $1.5h$ and $3h$, while for $D = 1h$, the requests are served in $15 - 50$ minutes. In Figure 4.7 we also observe the time of the day dependency, with lowest delays during the peak activity hours. This means that the proposed algorithms offer best offloading performance during the hours when a 3G network is most heavily loaded.

4.4.5 Real-Time vs. Delay-Tolerant Offloading

Real-time offloading requires 3-4 times more Wi-Fi cells than the delay-tolerant HotZones algorithm. It is interesting to compare the offloading potential of the RT Wi-Fi Zones with our delay-tolerant HotZones algorithm. In order to perform this comparison (in addition to the analyzed setup with 30 HotZones), we run the HotZones algorithm with 60, 120, 240, 480 and 960 Wi-Fi covered cells. We do it for the values of the permitted delay $D = 0h$, $1h$, $6h$ and $24h$.

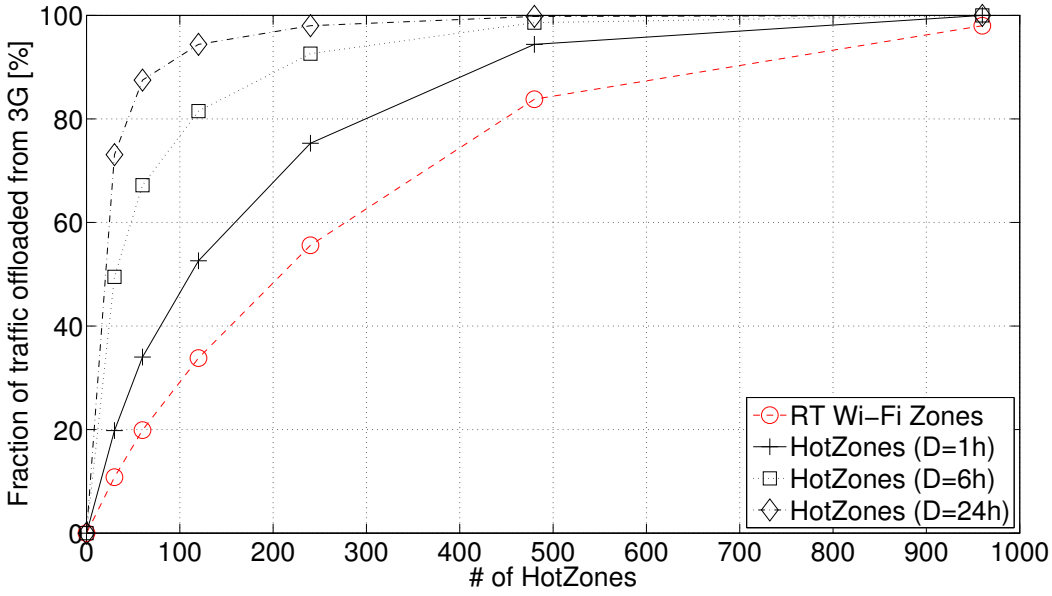


Figure 4.8: Offloaded traffic as a function of the number of HotZones.

On Figure 4.8, we can see that for $D = 6h$, covering only 10% of cells with Wi-Fi, results in offloading of 80% of traffic. In order to offload the same amount of traffic with $D = 0h$, an operator has to cover four times more cells with Wi-Fi. Similarly, the HotZones algorithm permits offloading of more than 90% of traffic with only 20% of Wi-Fi covered cells, while the RT Wi-Fi Zones require coverage of more than 70% of cells for a similar effect. This significant quantitative improvement, brought by prediction and delay tolerance in the HotZones algorithm, is even more valuable knowing that on average the delays are much lower than D.

4.4.6 MixZones Selection as a Compromise

MixZones selection is a compromise: Impossible to maximize both offloading and energy efficiency. When selecting the number of MixZones (*i.e.,* the algorithm parameter

P_{thresh}), we are guided by energy efficiency. The value $P_{thresh} = 0.8$ is most energy saving and it offers a solid offloading performance. Nevertheless, one can opt for another criterion when choosing the value of P_{thres}. On Figure 4.9, we plot the amounts of offloaded traffic and energy saved for the values $P_{thresh} = 0.2$, 0.5, 0.8 and 0.9. For these values we get 732, 590, 225 and 131 MixZone cells respectively.

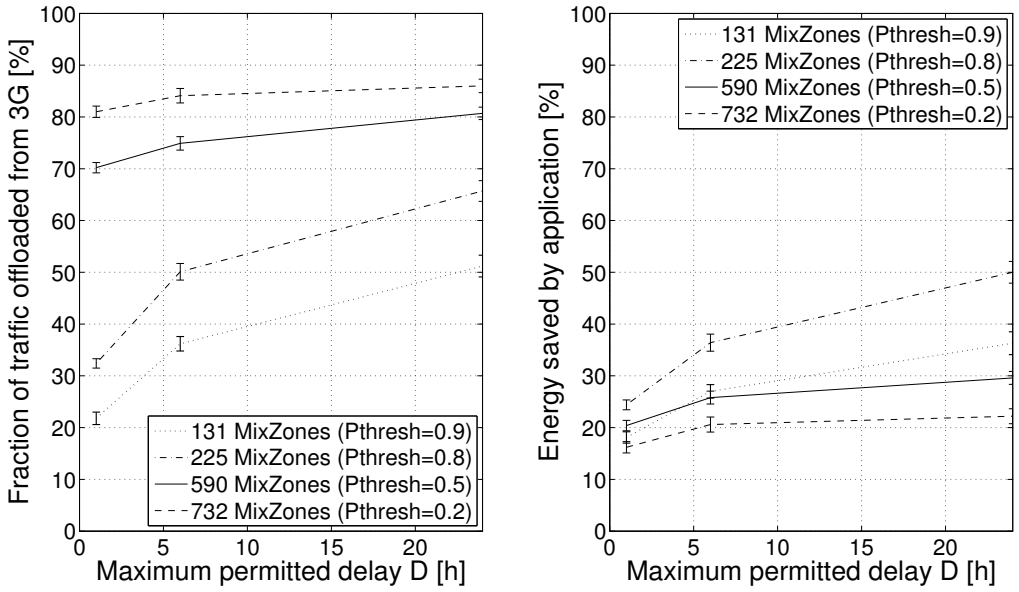


Figure 4.9: The offloaded traffic and saved energy as a function of the number of $P_{threshold}$ (*i.e.,* the number of MixZones).

We see that although the value $P_{thresh} = 0.8$ guarantees most energy saving, more traffic gets offloaded for $P_{thresh} = 0.2$ and 0.5. On the other hand for $P_{thresh} = 0.9$ both offloading and energy saving deteriorate. This can be interpreted in the following way. With the decrease of P_{thresh} (increase in the number of MixZones), the number of Wi-Fi transfers increases. However, these new MixZones have lower probability of meeting between users, which results in the increased number of Wi-Fi scanning events without data transfers. This decreases the energy efficiency. On the other hand, the increase of P_{thresh} beyond the value of 0.8, reduces both, the amount of offloaded traffic and the energy saving, due to too few cells that satisfy this condition.

4.5 Conclusions

In this chapter we study the use of prediction and opportunistic bandwidth for offloading of large, socially recommended contents from 3G networks. We show that the two algorithms we design enable users to trade delay for energy and easily reduce battery consumption coming from 3G transfers of delay-tolerant content for 50%. We show that the real-time offloading requires Wi-Fi coverage of 3 to 4 times more cells, than our delay-tolerant algorithm. We find that both algorithms have lowest delay during the peak hours, when offloading is most needed. We also demonstrate how operators can benefit the collected data to offer cloud solutions, appealing to users (extending battery lifetime) and to the operators (load balancing between orthogonal technologies).

We believe that performance evaluation of the algorithms using a realistic application and a large data set is a great contribution on its own. It helps community get better idea of the performance of a large scale delay-tolerant application in the context of a mobile network. It also allows us to gain insight into the possibilities of orthogonal 3G offloading, which is a topic that is likely to become increasingly important in the days to come.

Finally, covering a cell with access points (to create a HotZone) incurs certain costs for the operator, which are not considered in this chapter. However, such a coverage could be facilitated (and the cost reduced) by the use of existing operator-owned wireless routers, which provide Internet access at home to operator's customers. In order to provide fairness, the scheme proposed in [97] could be extended to mobile subscribers.

Part II

Twitter in the Air: Beyond Contact-Based Simulations

Chapter 5

Performance of an Opportunistic Network Through a Real Application

Many performance studies of various facets of opportunistic networks are based on simulation with contact traces. They are used for estimation of fundamental networking measures, such as delay or delivery ratio [3, 2]. They are also used for the design of caching and forwarding strategies [98, 99] and in the studies that address the effects of adding infrastructure to an opportunistic network [22]. However, the findings of these studies are practically never validated using live experimental deployments and real applications.

Although it is intuitive that contacts between users are one of the key factors to take into account when modeling information propagation in an opportunistic network, contact traces have certain limitations. By default, contact-based studies do not address the limited transmission bandwidth [1, 2]. Traffic generation is artificial or obtained from a distribution [16, 25]. Certain technology limitations, such as the inability of Bluetooth to concurrently discover and send data are ignored. In addition to this, contact-based studies often assume infinite sizes of users' caches and data exchanges without prioritization [3]. This, coupled with the absence of a model for the limited transmission bandwidth, can lead to simulations of unrealistic data exchanges.

In spite of the obvious need to quantify the effects of these approximations, little effort has been invested in justifying the perpetual use of contact traces for the analysis and simulation in the area of opportunistic networks. In other words, little evidence confirms that values, obtained from the simulations on contact data sets, accurately describe the performance of opportunistic applications. This situation can be explained, to a large extent, by the high cost

and complexity of the real application deployments. They normally require the implementation of several system components, the availability of a significant number of mobile devices and the participation of a non-negligible number of volunteers.

In order to fill this void and gain insight into the performance of an opportunistic network perceived through a real application, we design and implement a system that enables mobile users to use one of the most popular social networking applications (Twitter) without a permanent data connection, by leveraging sporadic Bluetooth connectivity with other users. We then run an experiment with 50 users, during 2.5 weeks in order to collect contact traces and the application data that can be compared with the results of the contact-based simulations. Our goal is to find out whether contacts are sufficient to capture the performance of an opportunistic network, with or without a backbone component and if so, how to best use them to achieve this.

Instead of inventing a new application for this purpose, we deliberately decided to extend an existing web application - Twitter - to the intermittently connected opportunistic space. This significantly simplifies the bootstrapping phase (exploits an already established user base and relationships between users), shortens the learning curve of the experiment participants and allows them to keep their existing habits and use the application in a more natural way.

The choice of Twitter also allows us to cover several realistic use cases. For example, it is quite expensive for roaming users to synchronize their mobile applications with the Internet on foreign networks. However, for a broad set of applications, such as e-mail, Twitter, Facebook and other social-networking apps, the synchronization may not be needed in real time. An opportunistic Twitter application, with occasional access to data sinks that provide Internet connectivity, might deliver tweets with acceptable delays.

Another example where an opportunistic Twitter application (accompanied with a few points of interconnection with the Internet) can be of great help is deliberate shutdowns of telecommunication networks during protests. Internet blackouts primarily target Twitter and other social networks, with the goal of preventing information propagation. Solutions, such as voice-to-tweet software provided by Google and Twitter, can allow users to tweet using voice [100]. Nevertheless, when the mobile phone service is down, the opportunistic communication, supported by a few satellite Internet connections, remains the only option [101].

The main contribution of this chapter is that we show how the common practice of ignoring certain factors in the contact-based studies of opportunistic networks significantly affects important performance metrics. Specifically, we show that the contact-based simulations overestimate the delivery ratios up to 30%, while the estimated delays are 2-3 times lower than the experimental values. Further, we demonstrate how the default assumption in the contact-based

simulations about the unlimited cache sizes completely alters conclusions about the utility of a backbone in an opportunistic network. We verify the robustness of these conclusions by rotating three different caching strategies, ranging from extremely selfish to altruistic.

In addition to this, we show that the statistical analysis of the weighted contact graph can be a viable alternative to the contact-based simulation, when it comes to capturing certain aspects of opportunistic network performance. Namely, we find a strong dependency between a user centrality measure in this graph and the perceived delivery ratio and we fit a simple curve to this dependency. This allows us to predict users' delivery ratios based only on the graph extracted from the contact trace. We show that this dependency persists when a backbone is added to the network, meaning that it can be used to estimate the effects of adding infrastructure to an opportunistic network.

From the systems aspect, in this chapter we give a comprehensive insight into the performance of a medium-sized opportunistic application and, to a lesser extent, into users' reactions to it. It highlights the most important design choices needed to extend an existing web application to the world of intermittently connected devices, such as our proxy server, used for secure synchronization with an existing web application.

This chapter is organized as follows. After presenting the related work in Section 5.1, we describe our opportunistic Twitter application and the experimental setup in Section 5.2. We introduce the notation and metrics used in the rest of the chapter in Section 5.3. In Section 5.4, we analyze certain properties of the data sets obtained from the experiment. Then, in Section 5.5, we compare the experimentally obtained application performance with the results of the simulations performed on the collected contact traces. This comparison allows us to pinpoint the common traps and pitfalls of the simulation based approach. In addition, we use the obtained data to gain insight into users' reaction to the observed performance (albeit to a limited degree) and to compute the costs of cooperation associated with our architecture and the opportunistic nature of the evaluated application. Finally, in Section 5.6 we show that a statistical analysis of the contact graph can predict certain aspects of network performance better than the contact based simulations.

5.1 Related Work

The validation of simulation results with measurements has been used before to evaluate the accuracy and determine the level of fidelity of simulation models [102], [103]. However, to the best of our knowledge, the work presented in this chapter is the first effort to study the

limitations of contact-based simulations in opportunistic networks. This is somewhat surprising, given the variety of topics and proposals validated using the contact-based simulations. A possible explanation can be sought in the cost, scale and complexity of the experimental setup, needed for such a study. Although the first of its kind, this study is closely related to a large body of work that addresses various aspects of opportunistic communication by using contact data sets. It concerns contact-based evaluations of caching and replication schemes [98], validations of forwarding protocols [99] and studies of content dissemination in urban environment [2, 16].

The work presented in this chapter is also closely related to the studies of the effects of a network backbone on opportunistic communication. Initially, these studies relied exclusively on contact traces [22], [3]. In [3], the authors perform extensive simulations using Bluetooth contacts, in order to quantify the effects of the opportunistic and backbone components on a delay tolerant network. They conclude that backbone brings only marginal improvements to opportunistic communication. The UMass DieselNet testbed addressed a similar topic, but the Wi-Fi equipped buses exchanged traffic (obtained from the Poisson distribution). The authors observe higher utility of the backbone component [25]. The work presented in the rest of this chapter permits to reveal that much of this discrepancy, in the observed backbone-induced improvement, comes from a common assumption in contact-based simulations about the infinite cache sizes.

Leveraging statistical properties of graphs that represent user relations has also been considered before for prediction of various performance measures and for protocol design. In [104], the authors propose the creation of a community content distribution network that would rely on "familiar strangers" (*i.e.,* collocated individuals with whom only limited interaction exists). Ioannidis et al. apply a similar approach [28]. They use so-called "weak ties" (*i.e.,* relationships with people outside the narrow social circles) to improve the dissemination of content updates over a mobile social network. In [27], the authors use centrality and communities obtained from a social graph for the design of a new forwarding algorithm. Finally in [105] properties of a social graph (extracted from users' activities) are used to predict customer churn in a cellular network.

5.2 Experiment Setup

We designed our experiment with two main goals in mind: (i) To collect the application data from which important performance metrics can be extracted and (ii) to collect the contact traces

that can be used in discrete event simulations. This allows us to compare the experimentally obtained results with the values obtained through simulation on contact traces, collected during the same experiment.

5.2.1 Experiment Scenario

In our experiment, we use the scenario of roaming users as the running example (although the scenario itself is not essential for the results of the study). We assume a mixed population, composed of visitors (Roaming Users) and users in their home networks (Home Users), at the university campus site. The policy restrictions often prevent Roaming Users (RUs) from connecting to the Internet via the campus WLAN (this is also the case at the EPFL campus). Thus, we assume that they prefer using free opportunistic applications to paying high roaming fees for data synchronization via the regular client applications.

As it is difficult to involve real visitors (in sufficient numbers) in a rather long experiment, we chose 50 volunteers to represent the RUs. While fully aware that the mobility of real roaming users can be somewhat different, we find that our experiment participants share certain mobility properties with campus visitors. As explained later in this section, about half of the participants are master students who followed courses in the classrooms where winter schools are organized (only for visitors). Also, all participants normally have lunch at the same places where visitors are likely to have lunch or coffee.

Home Users (students/faculty) are typically in majority. They have laptops with free access to the campus WLAN, and/or inexpensive data plans with mobile operators. We assume some of them are cooperative and willing to run a piece of software on their devices, helping Roaming Users deliver their tweets to the Internet and receive the tweets of the people they follow. Creating a significant Home User population for the purposes of the experiment (in addition to the Roaming User population we had to recruit) would require substantial financial and human resources. Thus, we resort to an abstraction. We place ten Linux laptops in popular places around the university campus (restaurants, computer rooms, coffee shops, libraries, etc.). We refer to these machines as Home User Equivalents (HUEs). We believe this is a good approximation, as (i) these are the locations where Home Users (with their cell phones and laptops) can be found during the day, (ii) the range of the Bluetooth dongles plugged into HUEs matches the Bluetooth range of cell phones and laptops ($\sim 10m$), and (iii) the set of functions handled by the HUEs is very limited, which means that the code can easily run on any piece of hardware (smartphones, laptops, etc.).

5.2.2 System Architecture

Our experimental setup consists of three main components (Figure 5.1): (i) Roaming Users (RUs) with the opportunistic Twitter application running on their phones, (ii) Home User Equivalents (HUEs) that serve as interconnection points between the opportunistic space and the Internet, and (iii) our proxy server, which is in charge of communication with the HUEs on the front-end and synchronization with Twitter servers on the back-end.

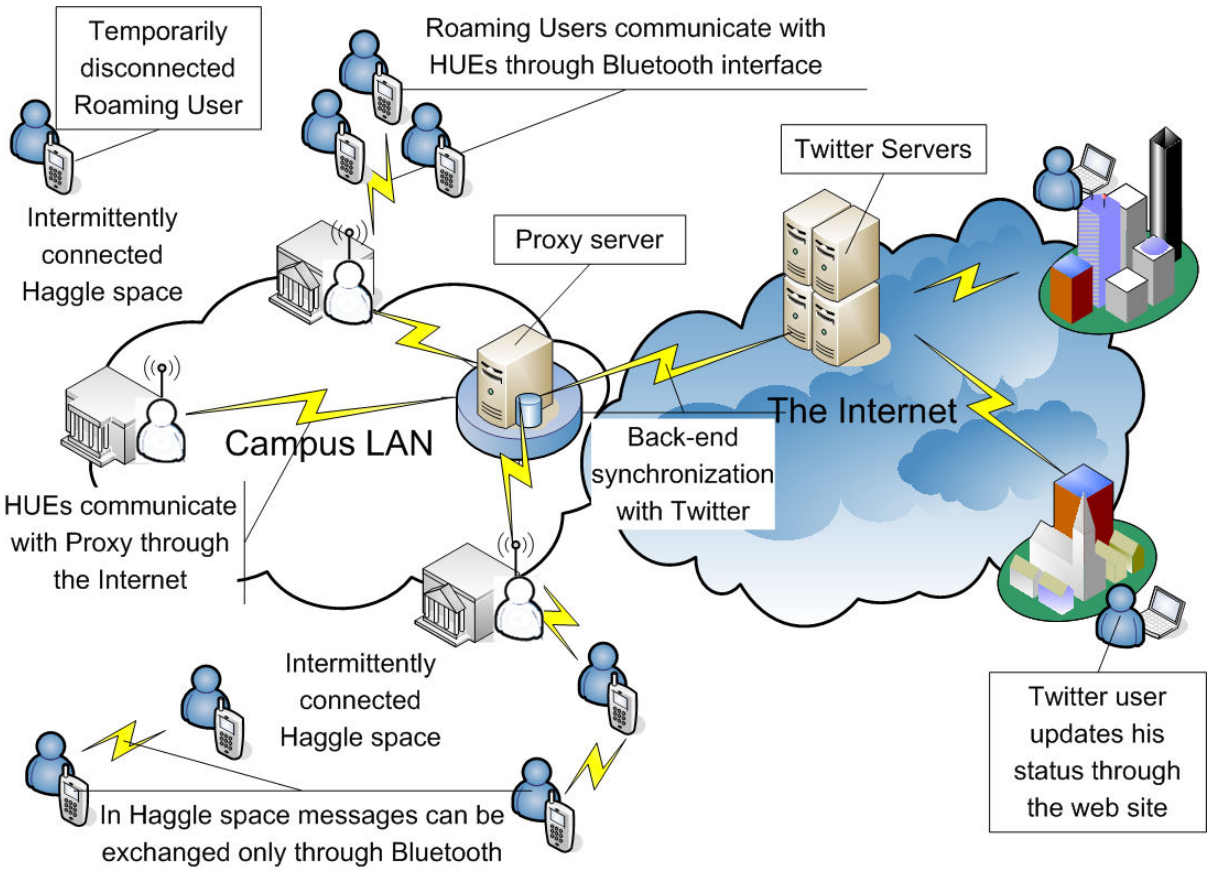


Figure 5.1: The system comprises three major components: (i) Proxy server, (ii) Home User Equivalents (HUEs), and (iii) the phones carried by Roaming Users (RUs). Proxy server communicates with Twitter servers at the back-end and with the HUEs at the front-end. HUEs provide Internet connectivity to RUs.

5.2.3 Opportunistic Twitter Application

Opportunistic Twitter is a mobile Twitter application we developed for the publicly available Haggle publish/subscribe framework [106]. It leverages intermittent Bluetooth connectivity for the exchange of messages with other devices running the Haggle framework. The

framework can work on top of several mobile and desktop operating systems (Windows, Android, Linux, Mac OS, etc.). In our experiment, the framework and the application run on HTC and Samsung Windows Mobile phones. Like most Twitter applications, our application allows to a user with a Twitter account to follow a set of other users or channels. As a result, messages ("tweets") created by these channels become visible in the user's message feed. So, every RU in our experiment has a group of other users/channels that he follows, as well as a group of followers that receive his updates. We refer to these relationships as the "Twitter following relationships". The tweets created by RUs, as well as the changes in the "Twitter following relationships," propagate through the network using a form of epidemic, as explained later in this section. Vibration informs users of message reception.

For the discovery of nearby users, all RUs and HUEs use the Bluetooth inquiry mechanism that allows them to find other Bluetooth devices within transmission range. Conducting inquiries consumes power, so one has to be moderate when setting the inquiry interval. Additionally, while inquiring, a device cannot answer other devices' inquiries, so performing frequent inquiries is not the best solution. On the other hand, choosing a too large interval results in missed discoveries (exchange opportunities). As users can recharge their phones on a daily basis, we choose a trade-off inquiry interval of 2 minutes. If a contact that was seen during the previous inquiry disappears during the following inquiry, but reappears again during the subsequent inquiry, we assume that the recorded contact was never broken.

5.2.4 Home User Equivalents as an Abstraction for Home Users

HUEs run a small application on top of Haggle framework and they have Internet connectivity. This allows to RUs to use them as sinks and have their tweets delivered to the Internet, *i.e.,* to their external followers around the world. HUEs can also fetch content from our proxy server and deliver tweets from the Internet to the RUs inside the campus.

The hardware that we use for the Home User Equivalents are Asus Eee PC mini-notebooks running Linux. They are equipped with Bluetooth dongles that have a radio range of approximately 10 meters, so in order to start a message exchange with a HUE, a Roaming User has to be physically close to it.

It is important to note that in spite of our cost-driven choice to use HUEs instead of HUs in the experiment, HUs are not envisioned as pieces of infrastructure. No infrastructure placement (access points of any kind) nor any assistance from mobile operators are required. HUs are conceived as local users with cheap Internet access, willing to share a part of their bandwidth

with roaming users. The reason why we chose to use mini-laptops placed in popular locations (and connected to the Internet) as an abstraction for Home Users is our conviction that this choice provides a good approximation of the places where Home Users can be typically found.

An alternative to this approach is to use a subset of the experiment participants as real mobile Home Users. We indeed implemented a proof of concept opportunistic Twitter application client for Android (which we dub HTweet) that allows for this. HTweet enables any opportunistic Twitter user to become a Home User, as soon as a data connection becomes available to him (for example when the user returns to his home cellular network). In other words, a user can become a gateway towards Twitter servers for other opportunistic Twitter users equipped only with Bluetooth, who can not or do not want to access the local network (for example due to high roaming costs). However, we opted not to use the HTweet and mobile Home Users during the experiment, as we believe a larger Home User population is needed in order for this approach to be viable.

5.2.5 Proxy Server

Our proxy server is a part of the system that resides between Twitter servers and HUEs. It is a component that is essential for the secure exchange of tweets between opportunistic Twitter users and Twitter servers (*i.e.,* secure user authentication), given the design of the Twitter API and the security architecture that we propose in Chapter 6. Thus, the proxy server is used for storing the data important for the operation of the system and for the post-experimental data mining. We implement it as a Java Web application running on the Apache Tomcat 6 server, which uses a MySQL database for data storage.

On the back-end (the interaction with Twitter servers), the proxy (i) passes to Twitter servers the tweets that arrive from HUEs and (ii) fetches from the Internet the tweets of interest to the experiment participants, by synchronizing the local copies of their accounts with their accounts on Twitter servers. In both cases Twitter servers require authentication of the users in question. The exact way in which this was handled by the proxy server at the time of the experiment and in which it can be done now (after the change of the Twitter API in late 2010) is explained in Chapter 6.

On the front-end (the exchange with the HUEs), the proxy server processes the messages received from the HUEs, it performs the database transactions and sends back the messages that need to be pushed into the opportunistic Haggle space.

5.2.6 Data Format

Messages exchanged in the Haggle ad hoc space (between RUs, or between RUs and HUEs) are in XML format. They contain application attributes (important for the operation of the system and data mining) and Haggle meta data. Several types of messages, each with different role, are used. Apart from regular tweets, there are messages that carry information about new channels being followed.

The HUEs and the proxy server communicate through the Internet using XML over HTTP, which allows for easy parsing on both ends.

5.2.7 Caching Strategies

Caching in RUs. Caching strategies (also called replication strategies) determine the channels that a user should store on the device and then forward. Note that *channels*, not packets, are the proper abstraction for Twitter traffic propagation, contrary to forwarding strategies in opportunistic networks. The reason is that users express their interests by choosing channels to follow. These interests remain relatively stable and they do not change on a packet-by-packet basis.

One can classify caching strategies according to how selfish they are: The more selfish a strategy is, the more preference it gives to channels that the user is interested in. In contrast, the more altruistic a strategy is, the more it prefers channels that are of interest to the rest of the community (network). The choice of strategy can affect network performance metrics.

In our experiment, we want to make sure that our conclusions are robust with respect to the choice of caching strategy, so we use three very different strategies. The first strategy is extremely selfish, storing only channels that the user is subscribed to; the second is extremely altruistic, preferentially storing channels that the user is *not* interested in; the third, which we refer to as proportional strategy (proportional to channel popularity), balances between the two extremes. More specifically, the third strategy [98] always stores the channels that a user is subscribed to and uses the remaining cache space for helping other channels. When two devices meet, each helped channel is a candidate for replacement, and each device performs the following operations: A locally helped channel c is selected uniformly at random among all locally helped channels, and a remote channel c' is selected uniformly at random among those remote channels that are not locally present. Then, the local channel c is dropped and replaced by the remote channel c' with probability $\min\{1, \frac{\beta_{c'}}{\beta_c}\}$, where β_c is the number of users following channel c.

Although considering an altruistic strategy can seem like a strange choice, it is important to understand that the caching strategy can be chosen by someone else, other than the application users. For instance, an application developer can intentionally add a dose of altruism in order to improve the overall performance.

We choose cache sizes for the RUs that we believe are commensurate with the parameters of our experiment, *e.g.,* , the number of users (devices), the amount of traffic that they generate, and the device hardware capabilities. Our objective is to examine the effect of a constrained cache size on the performance of the application. If the cache size is large, it will be practically infinite for the purposes of our experiment, so the results would not be representative for a larger network. We present results for cache sizes of 10 and 20 messages. The rationale is to be able to use the obtained results as best-effort indications of the performance in a larger scale deployment. Additionally, cached tweets are aged out after 8 hours, as we assume that older tweets are of no interest to Twitter users.

Caching in HUEs. Home User Equivalents (HUEs) have Internet connectivity. They can access all tweets available at the proxy in real time. However, keeping all tweets of interest to RUs in HUEs' caches is unwise. Downloading all these tweets to the local HUEs' caches, would increase the bandwidth cost for HUEs. Additionally, this approach has a scaling issue with the increase in number of RUs. Thus, we make the content available at HUEs adapted to the context, *i.e.,* to the interests of RUs in the vicinity of HUEs and other RUs that can be reached in the near future. HUEs have caches of 40 messages and they are refreshed upon reception of messages from RUs and messages pushed by the proxy.

5.2.8 Putting it All Together

Message Flow. The users create tweets that are forwarded among them in the following way: Upon a meeting between two users, messages in their caches are exchanged over Bluetooth. The Haggle pub/sub framework prioritizes message exchanges according to user interests: Messages of higher interest will be exchanged first, followed by the remaining messages in the cache of the other user's device. This prioritization is crucial when contacts are too short to exchange all messages of both caches. After the exchanges are over, the local caching strategy decides which messages, if any, should be dropped. To avoid transmitting messages that are then dropped, we align the Haggle prioritization with the caching strategy used at the time.

The HUEs are interconnection points between the Internet and the disconnected Haggle

space. The reception of a message from a RU triggers creation of an HTTP request by the HUE that is sent to the proxy through the Internet. The proxy processes the request, performs necessary transactions with the database and returns a set of messages ("tweets") as response. The HUE adds these messages to its local cache and makes them available to Haggle devices in its vicinity.

Experiment Population. Our RUs' population consisted of 50 people. Most of them received phones with the opportunistic Twitter application; some of them used their own phones. For the rest of the paper, we will be referring to our population of Roaming Users (RUs) also as *internal users*.

Many participants continued using their existing Twitter accounts. The others were free to choose the channels to follow. A followed channel can be either internal (content created by an internal user) or external (content created by an arbitrary Twitter user on the Internet, henceforth collectively called *external users (or channels)*). The social graph obtained from the "Twitter following relationships" shows that almost all internal users follow some internal and external channels. As the content created by external users is also propagated in our system we can, in a way, consider the external users as a part of the experiment.

5.3 Notation and Metrics

Let $\mathcal{N} = \{1, \ldots, N\}$ be the set of internal users, let $\mathcal{X} = \{1, \ldots, X\}$ be the set of external users, and let $\mathcal{F}_j \subseteq \mathcal{N} \cup \mathcal{X}$ be the set of users that user $j \in \mathcal{N}$ follows.

Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{N} \cup \mathcal{X}$ be arbitrary subsets of users. We use $\mathcal{M}_{\mathcal{A}\rightarrow}, M_{\mathcal{A}\rightarrow}$ for the set and number of messages generated by any user $i \in \mathcal{A}$; $\mathcal{M}_{\rightarrow\mathcal{B}}, M_{\rightarrow\mathcal{B}}$ for the set and number of messages delivered to any user $j \in \mathcal{B}$, and $\mathcal{M}_{\mathcal{A}\rightarrow\mathcal{B}} = \mathcal{M}_{\mathcal{A}\rightarrow} \cap \mathcal{M}_{\rightarrow\mathcal{B}}$. Only the messages generated by users that user j follows can ever be considered to be "delivered" to j, but not the messages that user j receives just to forward on behalf of others.

For an internal user $j \in \mathcal{N}$ and a message $m \in \mathcal{M}_{\rightarrow j}$, let D_j^m be the delivery delay of message m to user j. That is, D_j^m is the time elapsed between the generation of m at some user $i \in \mathcal{F}_j$ and the delivery of m to j.

For internal users $j \in \mathcal{N}$ we define the following metrics: The *delivery ratio* $R_j^{\mathcal{A}}$ from $\mathcal{A}$ to j is the fraction of messages generated by users in $\mathcal{A}$ and delivered to user j over the total

number of messages generated by users in $\mathcal{A}$ and destined for user j.

$$R_j^{\mathcal{A}} = \frac{M_{\mathcal{A}\to j}}{M_{\mathcal{A}\cap\mathcal{F}_j\to}}. \tag{5.1}$$

When $\mathcal{A} = \mathcal{F}_j$ we drop $\mathcal{A}$ from $R_j^{\mathcal{A}}$, and we simply call R_j the *delivery ratio*; $R_j^{\mathcal{N}}$ is the *internal delivery ratio*, and $R_j^{\mathcal{X}}$ is the *external delivery ratio*.

We define the *message delay* $D_j^{\mathcal{A}}$ from $\mathcal{A}$ to j as the average delay over all messages generated by users in $\mathcal{A}$ and delivered to user j.

$$D_j^{\mathcal{A}} = \frac{\sum_{m\in\mathcal{M}_{\mathcal{A}\to j}} D_j^m}{M_{\mathcal{A}\to j}} \tag{5.2}$$

Again, as with the delivery ratio, when $\mathcal{A} = \mathcal{F}_j$ we drop $\mathcal{A}$ from $D_j^{\mathcal{A}}$ and we call D_j the *message delay*; $D_j^{\mathcal{N}}$ is the *internal message delay*, and $D_j^{\mathcal{X}}$ is the *external message delay*. The last two measures are interesting because they represent the average reception delays perceived by user j for the messages created by internal and external users that user j follows.

We are also interested in evaluating the quality of the synchronization between the opportunistic part and the Internet part of the application. For this purpose, we treat the Proxy server as another user and measure its delivery ratio and message delay. We use the same two definitions as for mobile users, but we assume that the Proxy follows all internal users.

5.4 Obtained Data Sets

In this section we consider certain properties of the data sets acquired from our experiment that give us some insight into participants' behavior and activity during the experiment. These data sets are then used in Section 5.5 for the comparison between the system performance measured by the experiment and the performance obtained through post-experiment simulation on the collected contact traces.

As a result of the experiment we get two data sets: (i) the application metadata that we use to extract the fundamental performance metrics, such as delay and delivery ratio and (ii) the contact trace that we use in the trace driven simulations to obtain the same metrics from the collected contacts. Each of the three caching strategies applied at RUs is evaluated for two different cache sizes: 10 and 20 messages. This gives a total of six combinations, each of which is tested during two working days.

In the trace driven simulations that we perform after the experiment *we implement the same*

combinations of caching strategies and cache sizes. Each combination is simulated using the corresponding 2-day contact trace. Additionally, we simulate the case with infinite cache sizes, which often appears in the related body of work.

In our experiment, an average internal user (RU) follows 9 internal and 14 external channels ($>$ 600 external channels in total). The maximum number of internal and external channels followed by an internal user are 17 and 98, respectively. The most popular internal channel is followed by 18 internal users, while the most popular external channel has 8 internal followers.

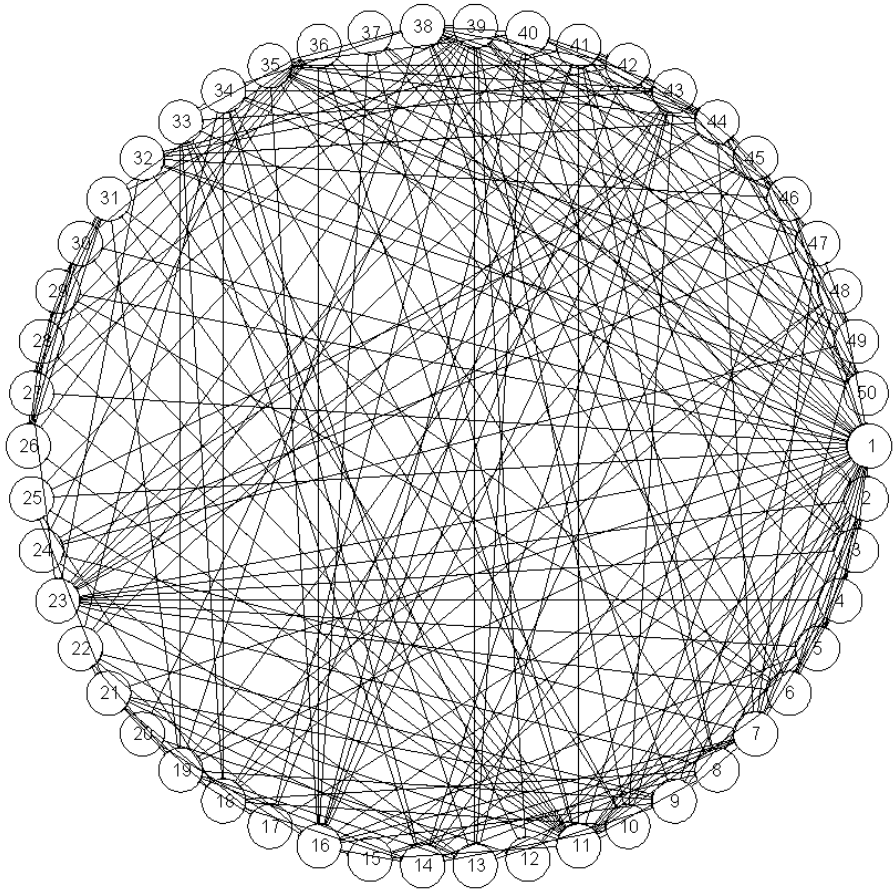


Figure 5.2: "Twitter following relationships" between internal users. Each vertex of the graph represents an internal user (experiment participant). An undirected edge between two users means that one of the two users follows the other one or that they mutually follow each other.

"Twitter following relationships" between internal users are shown in Figure 5.2. Each vertex of the graph represents an internal user. Each undirected edge between any two users i and j, where $i, j \in \{1, \ldots, 50\}$ signifies that user i follows user j; or user j follows user i; or users i and j mutually follow each other.

Figure 5.3 shows the total number of contacts that each of the 50 internal users and 10

HUEs have with other internal users and HUEs, during the six observed 2-day periods. We distinguish between the contacts with followed internal users, the contacts with other internal users and the contacts with HUEs. The total number of contacts varies depending on user ID and the day of the week. For example, the students of the Master's program have lectures and labs together on Thursdays and Fridays. Subfigures 1, 2, 4 and 6 (enumerated from top left to bottom right), which correspond to 2-day periods that contain either Thursday, Friday or both, clearly show more contacts (116, 98, 123 and 116 contacts per user per day, respectively) than subfigures 3 and 5 (56 and 59 contacts per user per day), which correspond to combinations of Mondays, Tuesdays and Wednesdays.

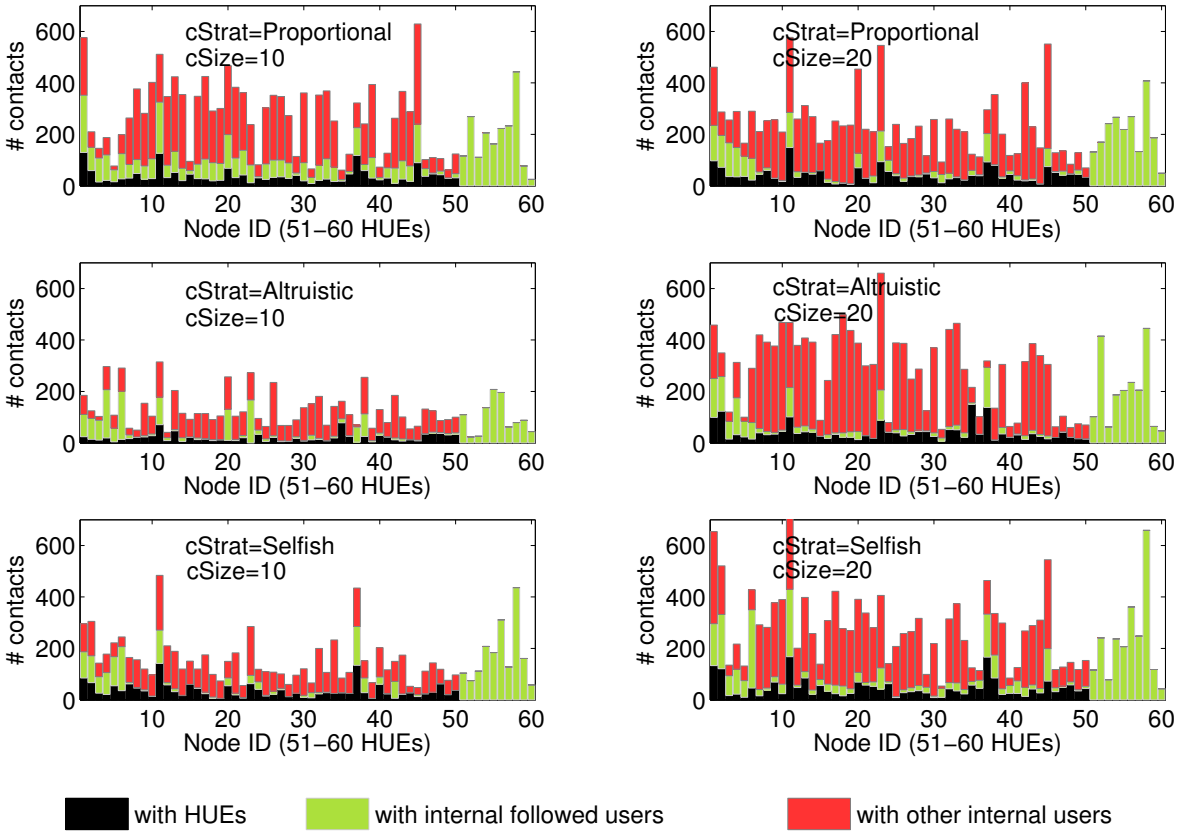


Figure 5.3: Total number of contacts experienced by internal users [$j = 1, .., 50$] and HUEs (gateways) [$j = 51, .., 60$] during 2-day evaluations of the 6 combinations of caching strategy $cStrat$ and cache size $cSize$.

The contact durations follow a similar pattern. For this reason, the comparison between the used caching strategies is not perfect, but it is also not the goal of our study. Obtaining identical contact patterns over all observed 2-day periods, with live and mobile experiment participants is hardly possible. Nevertheless, it is important to stress that the differences in

contact traces collected during different 2-day periods do not affect the conclusions of our comparison (presented in Section 5.5) between the experimentally obtained performance metric values and their counterparts obtained through contact-based simulation. This is because we always compare the experimental values for a given 2-day period with the simulation results acquired using the contact trace collected during the same period.

In Figure 5.4 we plot the complementary cumulative distribution function (CCDF) of the inter-contact times between the internal users and Home User Equivalents. It shows how often Roaming Users visit the popular locations within the campus where Home Users can be found. We view all HUEs as parts of the same backbone (as they all have access to the Internet) and we calculate inter-contact times with it for all internal users.

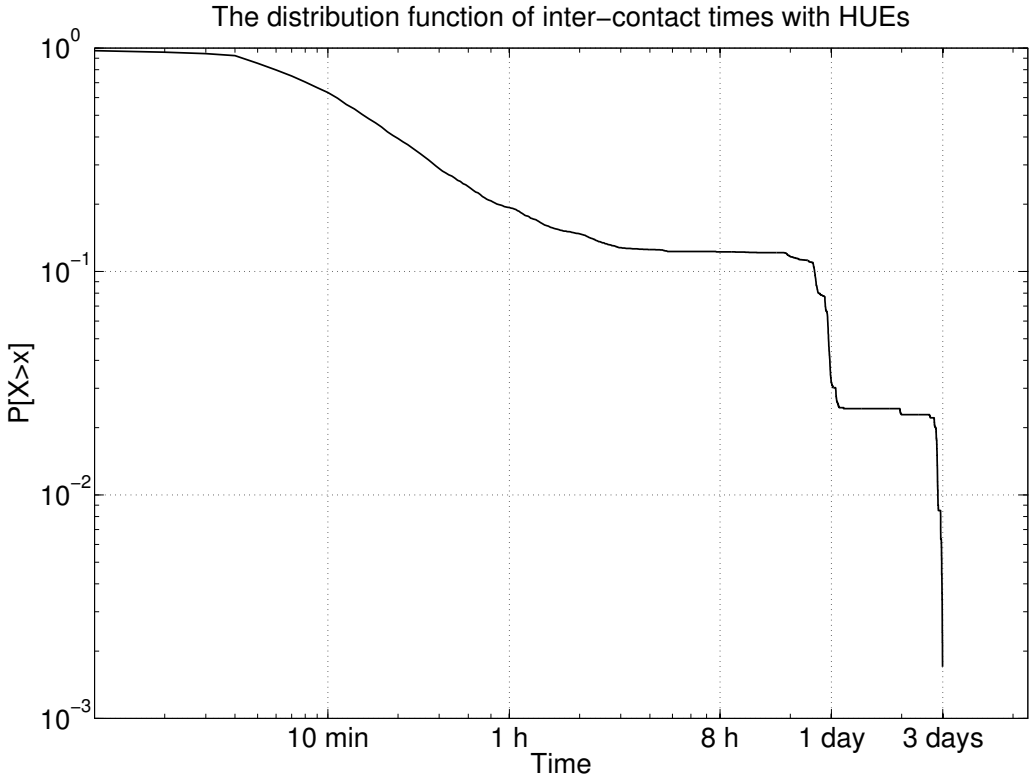


Figure 5.4: The complementary cumulative distribution function (CCDF) of users' inter-contact times with Home User Equivalents (HUEs) obtained for the whole duration of the experiment. All HUEs are perceived as parts of the same backbone (single entity), as they all have access to the Internet and the inter-contact times are computed accordingly.

The CCDF shown in Figure 5.4 is flat between 3 hours and 20 hours, which implies that it is more probable for a user to meet a HUE soon after the previous meeting. We also see that 80% of inter-contact times with Home User Equivalents is shorter than 50 minutes and only 3%

is longer than 24 hours. We observe two drops, at 20-25 hours and at 3 days, corresponding to meetings that happen once a day around the same time, and Friday meetings that happen again on Mondays.

In Figure 5.5 we show the complementary cumulative distribution function (CCDF) of the pair-wise inter-contact times among our experiment participants (internal users), calculated over the whole duration of the experiment. As this metric has been considered before, it is interesting to check if the findings of the previous contact-based studies hold in the case of the distribution function obtained from our contact trace. Indeed, we see the previously observed power-law and exponential decay properties of the function [24, 1].

Similarly to the function in Figure 5.4, we observe that 90% of the inter-contact times are shorter than 24 hours, which means that 90% of meetings were repeated (*i.e.,* the same users met again) within 24 hours. Only 1% of inter-contact times are longer than 1 week. Again, the flat region of the curve implies that users are more likely to meet again relatively soon after their previous meeting, and that the probability of a new meeting drops with the time elapsed since the last meeting.

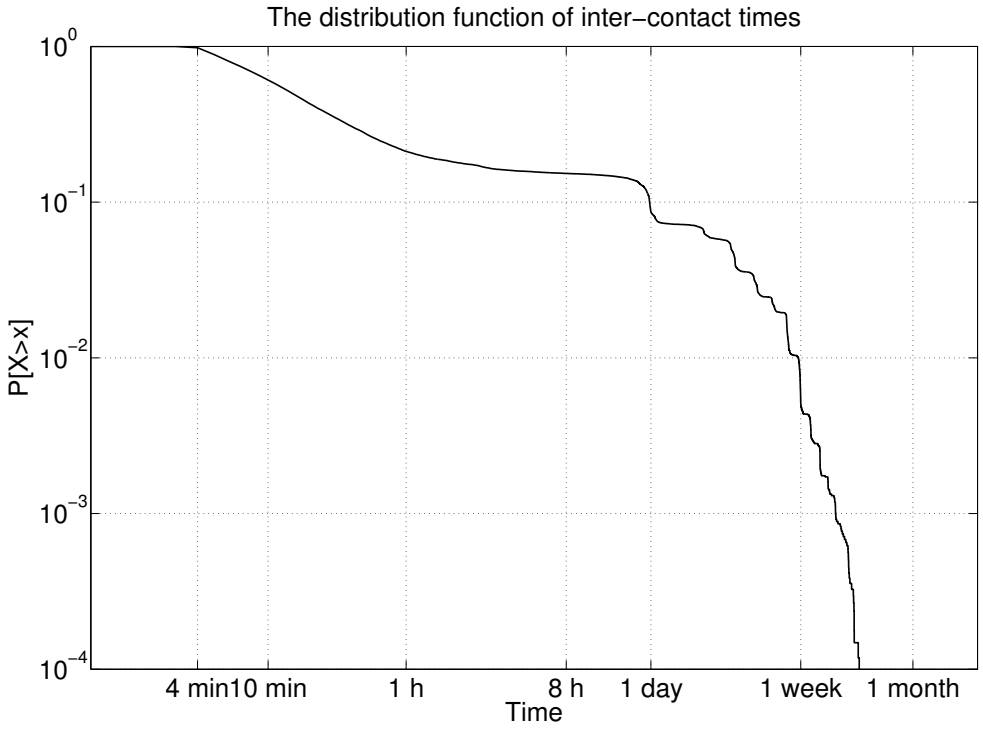


Figure 5.5: The complementary cumulative distribution function (CCDF) of the pair-wise inter-contact times among the experiment participants (internal users) obtained for the whole duration of the experiment.

Finally, before moving to Section 5.5, which contains the performance study of the evaluated opportunistic Twitter application, let us consider the quality of our HUEs placement. As previously explained, Home User Equivalents are placed in popular locations to represent Home Users with smartphones or laptops, who can normally be found in these places. Figure 5.6 shows the number of first copies of messages (as multiple copies can be created in the process of opportunistic multi-hop forwarding) delivered by each of the ten HUE to the Internet. We can see that each HUE delivered first copies of a non-negligible number of messages, with the most popular one delivering about three times as many messages as the least popular one. This means that all HUEs succeeded in serving their purpose as the points of interconnection between the intermittently connected opportunistic space and the internet. We also see that the internal users alone created 3010 tweets during 3 weeks of experiment.

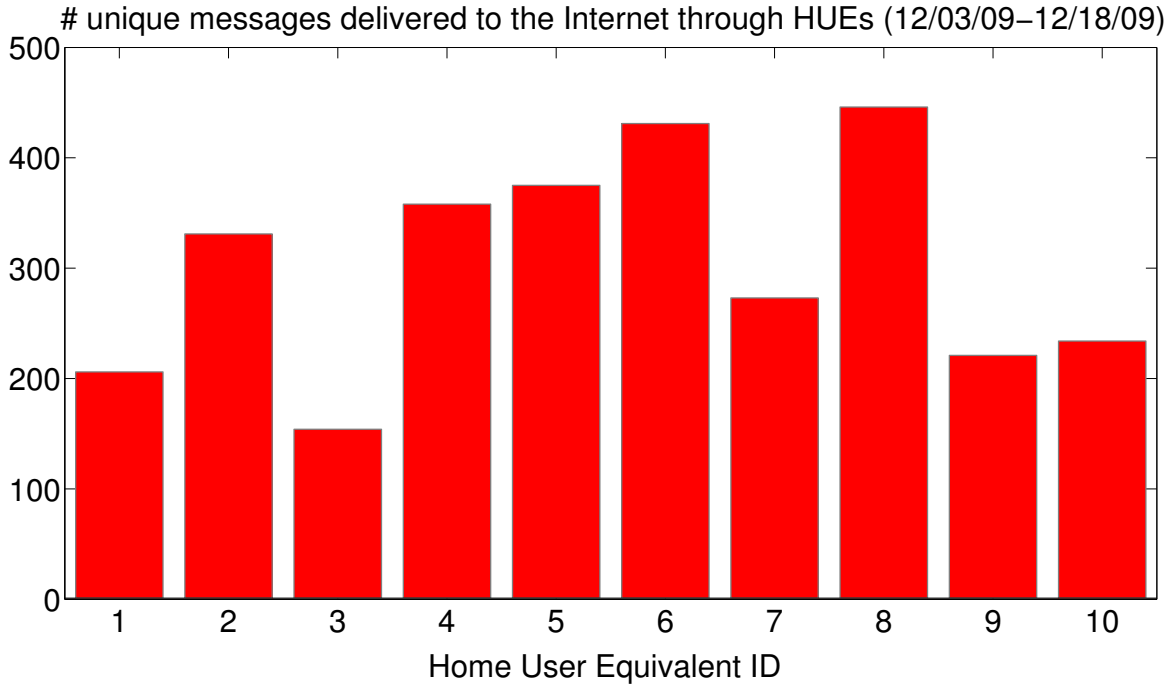


Figure 5.6: Number of unique internal messages first delivered to the Internet through HUEs $1-10$ from $12/03/09-12/18/09$.

5.5 Traps and Pitfalls of Contact-Based Simulation

The availability of the application metadata and contacts for the same experiment allows us to test the performance-related prediction accuracy of the commonly used contact-based simu-

lations. The rotation of caching strategies permits us to verify the robustness of our conclusions, *i.e.,* whether the conclusions persist for a range of different caching strategies.

We focus on two fundamental networking measures, namely, delay and delivery ratio, as defined in Section 5.3. In the case of both metrics we first analyze the values obtained from the experiment. We then compare these values with the corresponding values obtained from the contact-based simulations. Finally, we study the effects of adding a backbone to an opportunistic network, showing that as a rule, contact-based studies underestimate the impact of backbone, due to hidden assumptions.

5.5.1 Experimentally Obtained Delivery Ratios

Figure 5.7 shows the internal and external delivery ratios, $R_j^{\mathcal{N}}$ and $R_j^{\mathcal{X}}$ (as defined in Section 5.3), seen by internal users ($j = 1, ..., 50$) and by the proxy server ($j = 51$), during the observed evaluation periods. Each period of two working days corresponds to a combination

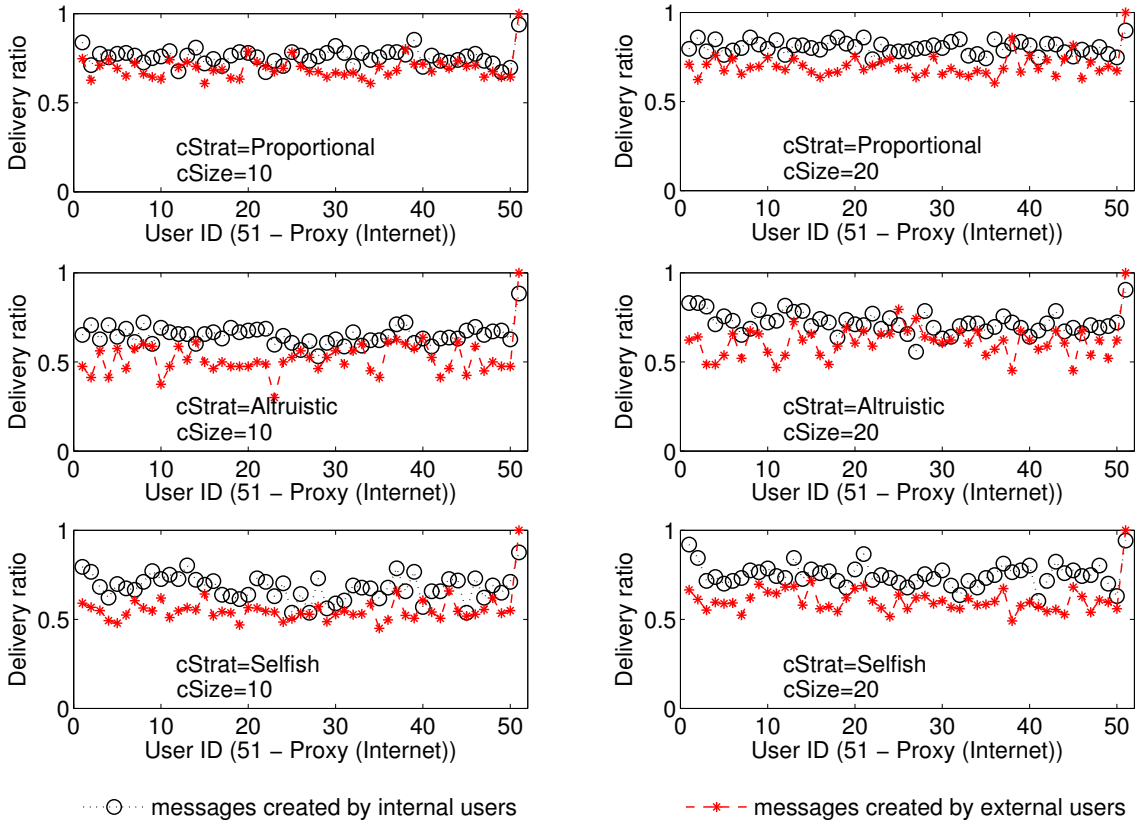


Figure 5.7: Internal and external delivery ratios, $R_j^{\mathcal{N}}$ and $R_j^{\mathcal{X}}$, seen by the internal users ($j = 1, ..., 50$) and by the proxy ($j = 51$). Every combination of caching strategy ($cStrat$) and cache size ($cSize$) was evaluated during 2 days.

of a caching strategy and a cache size. We see that proportional strategy performs on average 10-20% better for both evaluated cache sizes. We observe higher delivery ratios when the cache size is 20, regardless of the caching strategy. Finally, through the performance of user 51 (the proxy server), we see that almost all messages, created by internal users are delivered to Twitter web site.

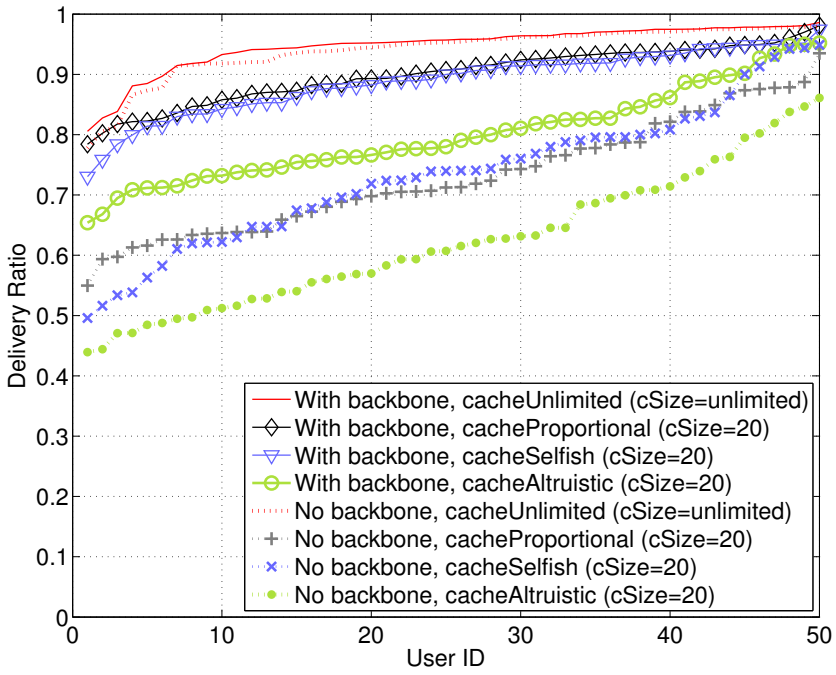
Figure 5.7 also shows that the external delivery ratio is lower than the internal. The reason is that the number of external channels is large (> 600) and there is only a limited overlap between channels followed by internal users. So, each cached external channel is useful to few internal users. Even if caches were full of external channels, there would still be channels that are not cached anywhere, thus making it difficult for the followers of these channels to receive them.

5.5.2 Contact Simulations Overestimate Delivery Ratios

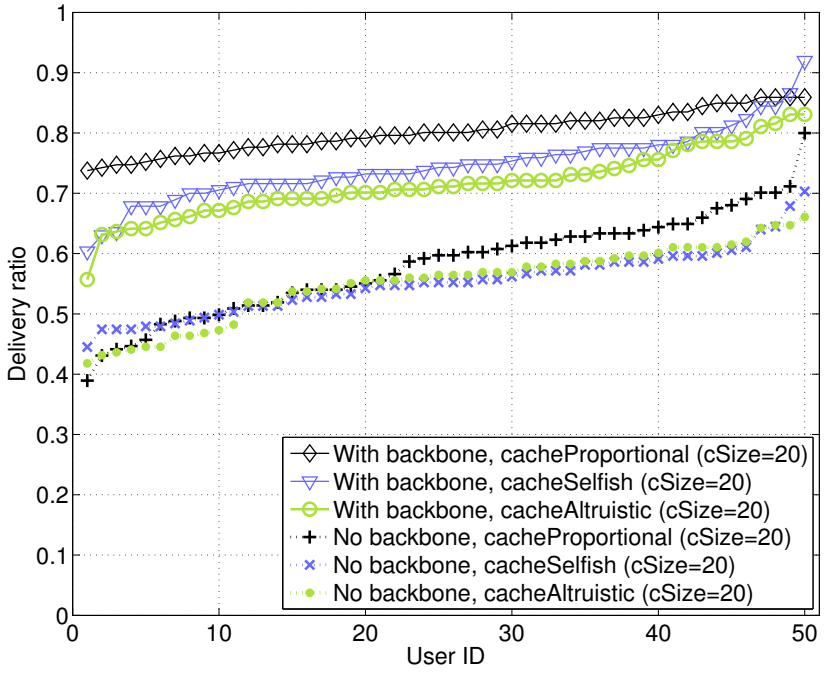
We now compare the experimentally obtained delivery ratios with the values acquired through the post-experiment simulation on the collected contact traces. Each combination of cache size and caching strategy is simulated using contacts collected during the period when this combination was used.

In Figure 5.8, the three full lines on the bottom subfigure correspond to delivery ratios perceived by experiment participants, for messages created by internal users that they follow and when using the three evaluated caching strategies with the cache size of 20 messages. The same unsorted values are shown on Figure 5.7, in the three subfigures on the right, which contain delivery ratios for the cache size of 20 messages. The top subfigure of Figure 5.8 contains the corresponding delivery ratios, obtained through the contact-based simulation with the same caching strategies and with the same cache size of 20 messages. In addition to this, the top subfigure also contains delivery ratios obtained through simulation with unlimited cache sizes, where users can cache all received messages (top full line). We simulate the case with unlimited cache sizes, because this is the most frequently used assumption in the existing literature [2], [3].

The two subfigures in Figure 5.8 allow us to draw the first two conclusions about the deficiencies of contact-based simulations. First, contact-based simulations overestimate delivery ratios. This is due to the fact that they fail to model the limited contact durations and transfer bandwidth, as well as the limitations of the used wireless technology. In other words, some recorded contacts do not result in transfers and some of them allow transfers of only a part of



(a) Delivery ratios - simulation



(b) Delivery ratios - experiment

Figure 5.8: Delivery ratios obtained from simulation and from the experiment for different caching strategies. The full lines correspond to the system with the backbone (HUEs, proxy server). The dotted lines describe the system in which only opportunistic internal users exist.

the available data. This is further confirmed in Section 5.5.5, where we compare the experimentally acquired delays with the delays obtained through the contact-based simulations. Second, assuming unlimited cache sizes always increases delivery ratios. For example, we see that this assumption increases delivery ratios for up to 30%, in comparison to the case with altruistic caching strategy and the cache size of 20 messages.

5.5.3 Misinterpreting the Importance of a Backbone

The data sets we collected during the experiment enable us to study the improvement that a backbone brings to opportunistic communication. This is possible, because the application metadata allows us to differentiate between the copies of a message that traversed the backbone (HUEs, proxy server) in the process of forwarding and those that reached their destinations using pure ad hoc forwarding among the experiment participants. By considering the former as lost, we calculate delivery ratios and delay in a hypothetical system without backbone connectivity.

As external messages cannot enter the system without the backbone, the metrics in the hypothetical system are about internal messages only. Similarly to the definition of R_j in Section 5.3, we define R'_j as the fraction of messages delivered to user j over the total number of messages destined for user j, in a system without a backbone.

The dotted lines in Figure 5.8 represent delivery ratios in the system without a backbone (HUEs, proxy server), for different caching strategies and for the cache size of 20 messages. Again, the contact-based simulation significantly overestimates delivery ratios (about 30% in the case of proportional caching strategy).

Figure 5.8 allows us to observe another trap of contact-based simulations. We see that in the case of limited cache sizes backbone brings significant improvement to delivery ratios. However, in the comprehensive simulation study in [3] the authors conclude that backbone brings only marginal improvement to delivery ratios. This conclusion is the result of an often hidden assumption in the contact based studies that cache sizes are infinite. Indeed, as we see in Figure 5.8, in the simulated case with unlimited cache sizes, backbone brings almost negligible improvement. This is due to the fact that a user with unlimited cache size can store much more information, so during a contact, he can provide almost as much data as a backbone.

5.5.4 Experimentally Obtained Delay

Figure 5.9 shows the internal and external message delays, $D_j^{\mathcal{N}}$ and $D_j^{\mathcal{X}}$, observed by the internal users ($j = 1, .., 50$) and by the proxy server ($j = 51$). The average internal delay typically ranges from 100 to 140 minutes. The average external delay is higher. Intuitively, one would expect the external messages to reach their destinations faster, due to their availability at all HUEs soon after creation. Messages created by internal users, in contrast, experience a non-negligible delay before becoming available at HUEs, as we can see from the delay observed by the proxy ($j = 51$ in Figure 5.9). However, as we observe in our message log, some of the external messages created in different time zones are created during the night. This introduces delay, as there are very few or no internal users on the campus in the nighttime.

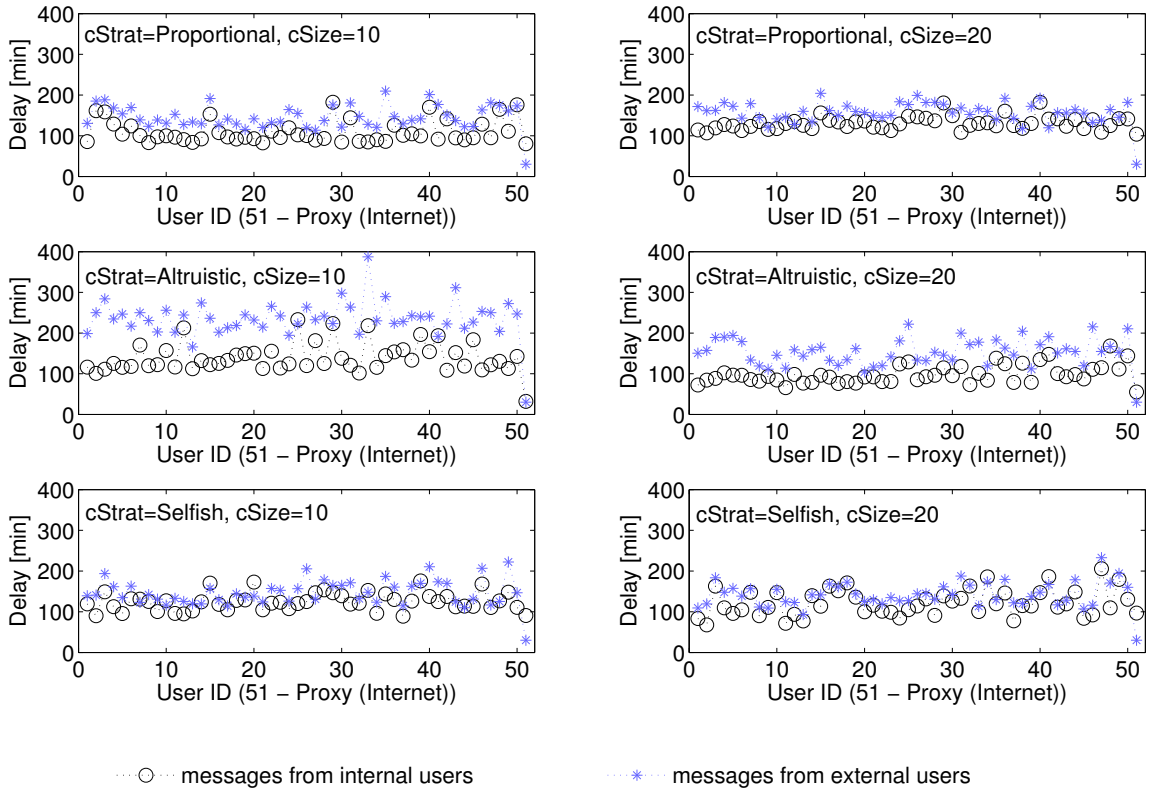
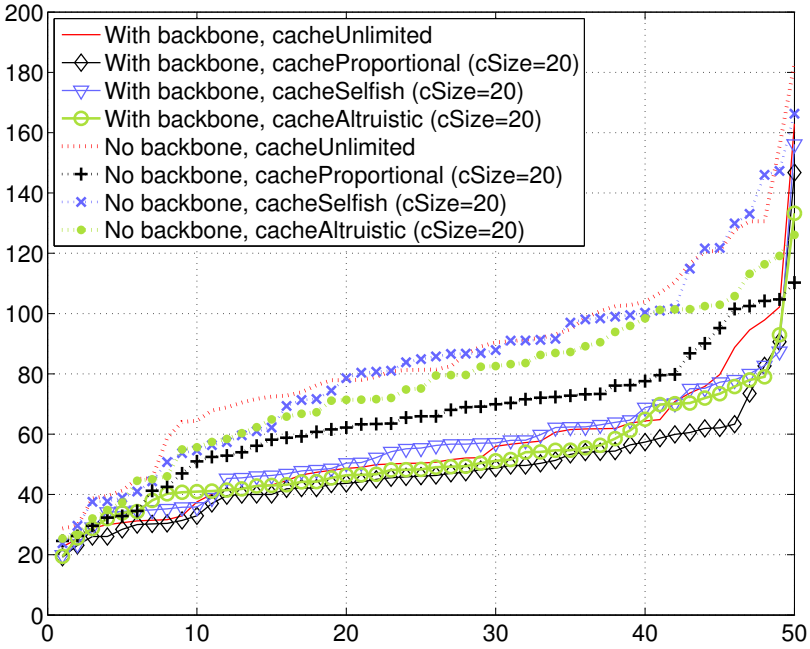
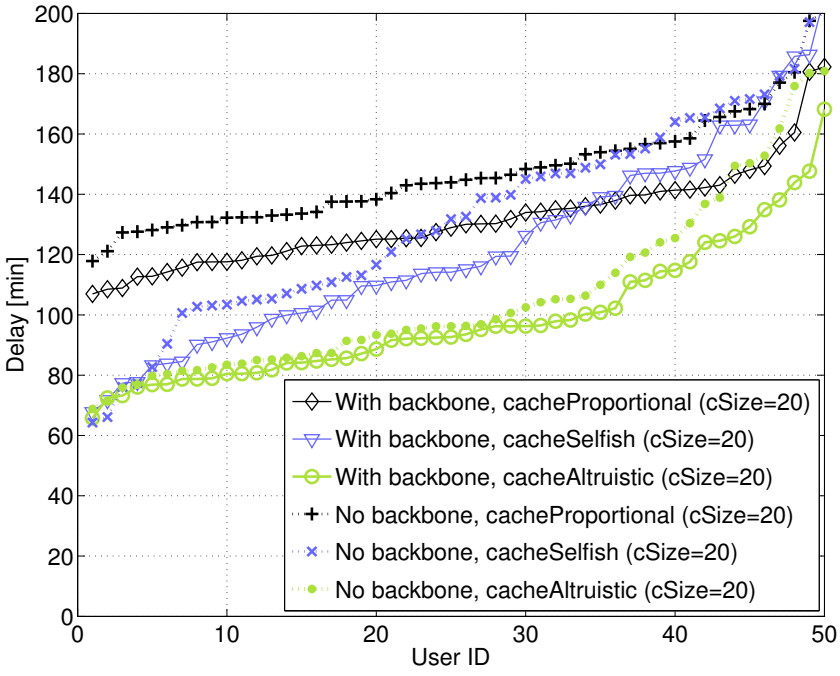


Figure 5.9: The average delay of received message observed by internal users. Every combination of caching strategy ($cStrat$) and cache size ($cSize$) was evaluated during 2 working days.



(a) Delays - simulation



(b) Delays - experiment

Figure 5.10: Delays obtained from the simulations and from the experiment for different caching strategies. The full lines correspond to the system with the backbone, while the dotted lines describe the system without the backbone. The case with unlimited caches is also simulated.

5.5.5 Contact-Based Simulation Underestimates Delay

In Figure 5.10 we plot delays obtained from the experiment and from the contact-based simulations, for the cases with and without backbone. We see that simulations give delays that are 2-3 times lower than the experimentally obtained delays. We inspect the contact trace and the application data and we observe that recorded contacts do not always result in message transfers. This means that limited transmission bandwidth, short contact durations and inability of Bluetooth to concurrently scan and send data prevents users from leveraging all transfer opportunities. As most of these limitations are not inherent only to Bluetooth, we conclude that delays obtained from contact simulations should be taken with a grain of salt, as they are too optimistic.

5.5.6 Delay from Users' Viewpoint

The recorded delays give us some insights into performance from the networking perspective. However, we would like to know more about users' perception of this performance and their reaction to it. More precisely, we would like to know what an average delay of 120 minutes represents from the users' viewpoint, *i.e.,* whether the user still finds this delayed content relevant and responds to it, or he just ignores it as a piece of obsolete information.

Twitter option called "@replies" allows us to find out more about this. When a Twitter user receives a tweet he wants to respond to, he can create an @reply message, by putting @ + the name of the creator of the original tweet in his reply. This helps us easily identify pairs containing original tweets and @replies to these tweets. We then record delays for the tweets whose reception led to the creation of @replies by the recipients and we plot the corresponding CCDF in Figure 5.11.

We can see from the figure that 60% of the tweets that receive an @reply are received with a delay inferior to 2h. However, 40% of the tweets that instigated the creation of an @reply message are received with a delay between 2 and 3h. This means that the recipients still find this non-negligible delay acceptable. In addition to this, we find that many of the @replies are threaded and parts of longer conversations (we also verify this by checking the message content), which means that the observed delays allow users to maintain longer message exchanges.

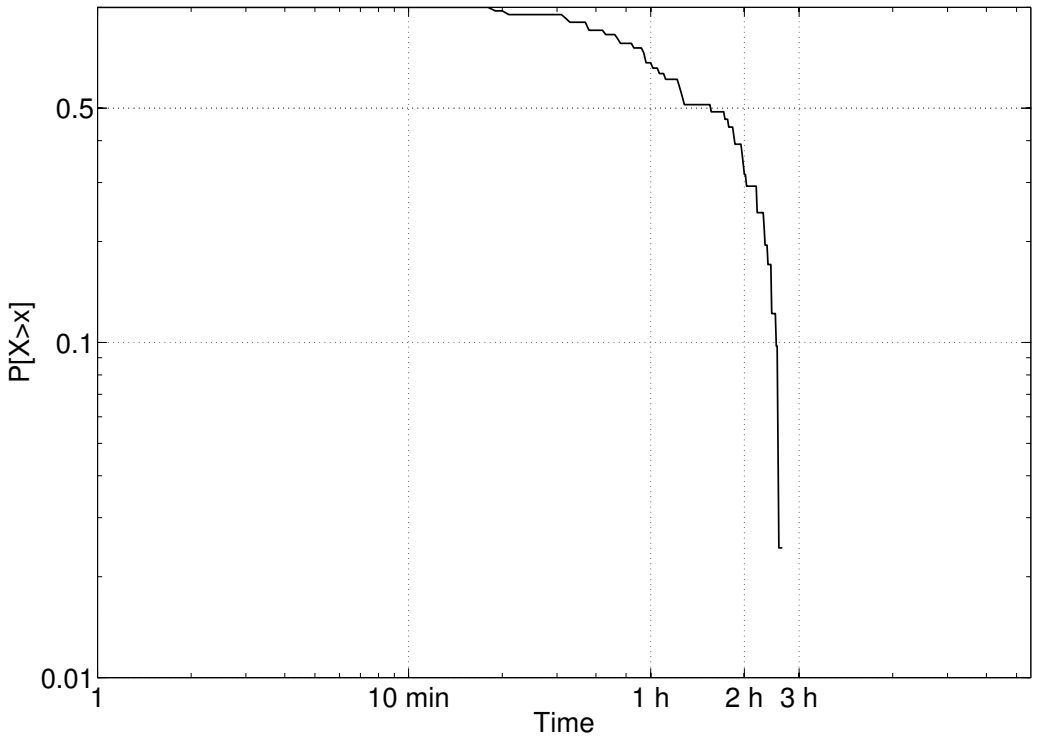


Figure 5.11: The distribution function of observed delays for the tweets whose reception led to the creation of @replies by the recipients.

5.5.7 Cooperation from Users' Viewpoint

The performance perceived by the RUs comes at a certain cost for the Home Users. In particular we refer to (i) the bandwidth cost, measured through the amount of data that needs to be uploaded/downloaded by a HU's device via $3G$, and (ii) the energy cost, especially in the case of smartphone devices. Using the data obtained from the experiment we compute these two aspects of the cooperation associated cost.

Bandwidth Cost. The bandwidth cost can be estimated by multiplying three quantities: the average number of daily contacts per device that result in data exchange (~ 25, as many registered contacts are too short to involve data exchange or they happen between the same devices soon after the contacts where data exchanges happened), the number of messages that are uploaded/downloaded upon a contact (~ 40) and the average size of a message ($\sim 1kB$). Thus, a cooperative HU is expected to upload/download $\sim 1MB$ on a daily basis. This corresponds to $\sim 3\%$ of a $1GB$ monthly data plan, which is probably not unbearable.

Energy Cost. Even more important than the consumed bandwidth is the energy cost. It can

be split into two components: Bluetooth associated cost and $3G$ transfer related cost. The daily Bluetooth associated cost comes from idle periods, scanning, and data transfers. We use the published Bluetooth consumption characteristics [51] to compute these individual energy tolls. Given the pattern of scanning used in our experiment ($\sim 10s$ every $2min$), the idle periods cost $22*3600s*0.01J/s = 792J$ per day; the scanning periods cost $2*3600s*0.12J/s = 864J$ per day; and the data transfers can be ignored for the daily amount of data our HUs send/receive.

The energy consumed by ~ 25 $3G$ uploads/downloads of $40kB$ of data is calculated (with the energy model of [48]) to be $25 * (0.025 * 40 + 3.5 + 0.62 * 12.5)J = 291J$ per day. The Bluetooth and 3G energy costs add up to a total daily energy consumption of $\sim 1947J$.

For an average laptop battery ($\sim 130Wh = 468kJ$) the costs calculated above are negligible. For a recharged smartphone battery ($\sim 5.5Whours = 19.8kJ$), the costs associated to providing Twitter access to roaming users amount to about 10% of battery power.

5.6 Using Contact Graph for Performance Prediction

In Section 5.5 we show that simulations on contact traces suffer from multiple drawbacks. A contact trace can also be analyzed using its statistical properties. The goal is the same, estimating the performance of an opportunistic network/application. In this section, we examine the usage of contact traces for the prediction of certain aspects of opportunistic network performance using an alternative approach. Instead of running simulations, we focus on the properties of the weighted contact graph. What makes this approach possible is again the availability of the experimentally obtained metric values and contact traces for the same experiment.

5.6.1 Closeness Centrality Predicts Delivery Ratio

We apply the following approach: to represent the contacts among users, we define the *contact graph* as an undirected weighted complete graph $G_{con} = (\mathcal{N} \cup \{I\}, E_{con})$. The vertex set comprises the internal users and the vertex I representing the infrastructure. As the graph is complete, the edge set E_{con} comprises all unordered pairs of vertices. The weight of the edge $ij \in E_{con}$ is equal to $w_{ij} = \frac{1}{c_{ij}^{\lambda}}$, where c_{ij} is the number of contacts between users i and j, and λ is a real number constant.

In the graph G_{con}, we denote by $d_{ij}(\lambda)$ the shortest path distance between i and j. The

average shortest distance $d_i(\lambda)$ of a node i (other than I) to all other nodes in the graph is

$$d_i(\lambda) = \frac{\sum_{j \in \mathcal{N} \setminus \{i\} \cup \{I\}} d_{ij}(\lambda)}{N}, \tag{5.3}$$

also called *closeness centrality* in the social network literature [107]. The lower this quantity is, the more connected a node is. We find a noticeable dependency between the delivery ratio R_i of a node i and the node's closeness centrality d_i. In particular, the following curve fits the data well:

$$R_i = \frac{1}{1 + k d_i(\lambda)}, \lambda = 0.95, \tag{5.4}$$

where k is a constant that depends on the caching strategy (discussed in Section 5.6.2).

Other centrality measures that we tested, namely degree centrality, eigenvector centrality and betweenness centrality, result in weaker dependency. By applying a similar approach to the social graph shown in Figure 5.2, we find no dependency between delivery ratio (or delay) and centrality measures in this graph.

5.6.2 The Curve Fitting Details

The weight exponent λ can change the relative importance of small and large edge weights. The weight of a path p is the sum of the weights of its edges $e_1, e_2, ..., e_l$:

$$w(p) = \frac{1}{c_{e_1}^{\lambda}} + \frac{1}{c_{e_2}^{\lambda}} + \ldots + \frac{1}{c_{e_l}^{\lambda}}. \tag{5.5}$$

With a large positive value of λ, the edges with a small number of contacts dominate, whereas with a large negative value of λ, the edges with a large number of contacts dominate.

We choose $\lambda = 0.95$ because this value maximizes the mutual information between $d(\lambda)$ and R, viewed as discrete random variables. Intuitively, the mutual information of $d(\lambda)$ and R is high when the knowledge of one reduces our uncertainty about the other, which is desirable as we want to use d to predict R. The advantage of using mutual information as opposed to, for instance, correlation, is that mutual information is not biased by the relative values of the quantities involved. So, we see that, to maximize the predictive power of d for R, all edge weights should be treated with approximately equal importance. After choosing $\lambda = 0.95$, we do curve fitting to find the value of k that minimizes the sum of vertical distances. The values of k are always in the interval $[2.7, 3.5]$.

Knowing the dependency and using the curve helps one estimate a typical node's expected delivery ratio if one can estimate or guess a typical node's closeness centrality. Furthermore,

one can form an expectation about the effect of connecting to the backbone (thus changing nodes' closeness centralities) on the delivery ratio that network users will experience.

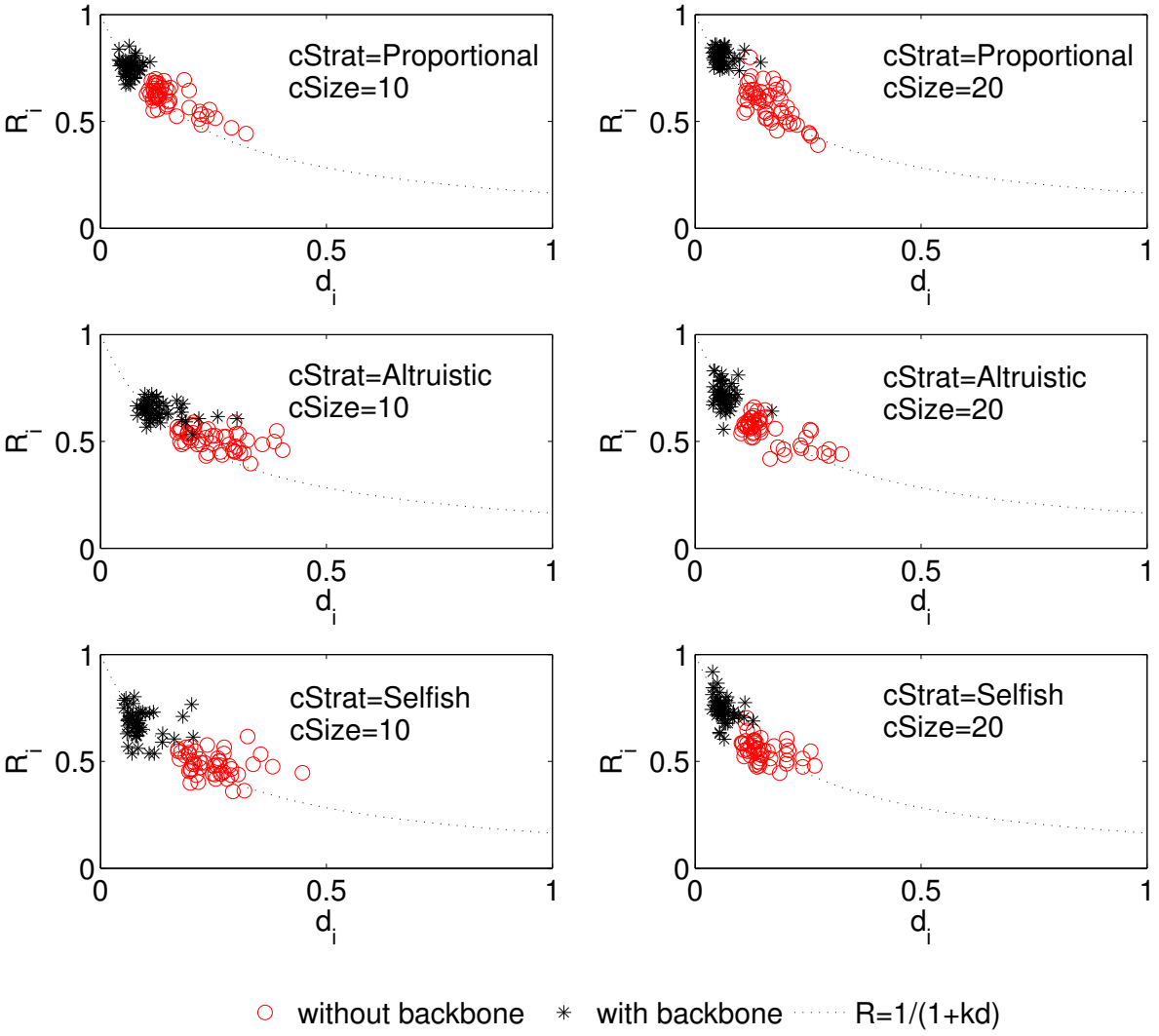


Figure 5.12: Dependency between delivery ratio R_i and closeness centrality d_i.

For $k = 3.1$, we plot the data and the curve in Figure 5.12. In every subfigure each user's R_i and d_i are plotted for the cases with and without backbone. We see that the $R - d$ dependency holds, not only across users within the same network topology, but persists even across qualitative changes in the topology.

Moreover, we see that the dependency exists, and the curve fits, *regardless* of the caching strategy used. This last point is important, because caching strategies affect delivery ratios. But the effect of the caching strategy can be included in the constant k, which is limited to a small

range of values. We conclude that a node's distance is a reliable indicator of its delivery ratio.

5.7 Conclusion

This chapter shows that studding live opportunistic applications can help us improve our understanding of the opportunistic network performance, as the results obtained through a live deployment differ in many respects from the simulation results. They also offer us a unique insight into some aspects of network performance, that are otherwise impossible to observe through the commonly used contact-based simulations.

Our experiment with a real application on top of an opportunistic network shows that the commonly ignored factors in simulation studies of these networks (such as technology limitations, limited contact durations, finite transmission bandwidth, etc.) lead to significant discrepancies between experimental and simulation values. All caching strategies and cache sizes, tested by 50 users during the 2.5 week experiment, unanimously confirm that contact-based *simulations overestimate network performance* (especially in the case of delay). This means that an effort should be made to include these missing factors in the future trace driven simulations.

In addition to this, we find that some commonly hidden assumptions, like the assumption about the infinite cache sizes, result in the overly pessimistic conclusions about the *utility of a backbone in an opportunistic network*. This is an interesting finding that could direct more attention towards hybrid networks, that include both, the opportunistic and the infrastructure component.

Finally, we show that a statistical treatment of the contact trace, offers a good prediction of certain performance aspects, namely delivery ratio. We show how the existence of a backbone increases the message delivery ratio by reducing user distances on the contact graph. The strong statistical dependency that we find (between node's centrality and delivery ratio) can help predict not only delivery ratios, but also the effect of adding a backbone to an opportunistic network.

Part III

Security for Opportunistic Applications and its Performance

Chapter 6

Security Architecture for Intermittently Connected Opportunistic Applications

The opportunistic Twitter application, described in Chapter 5, differs from the traditional (always connected) Twitter clients in several important ways. These differences are generic and they persist for a whole range of traditional clients and their opportunistic counterparts (which can be implemented following the design of our opportunistic Twitter application). The most important difference is the participation of intermediate users (hops) in the process of data forwarding. Security threats introduced by the multi-hop forwarding require us to rethink the security solutions used in the case of traditional application clients (designed exclusively for direct client-server communication). The goal is to provide comparable level of security to opportunistic clients that occasionally synchronize with web services and that rely on intermediate hops in the process of data forwarding.

At the same time, one has to bare in mind that our opportunistic Twitter clients do not form a completely autonomous network, as they occasionally access resources on the Internet. For this reason, the proposed security solutions for autonomous opportunistic networks cannot be used, as are, because they have to be compatible with the existing security APIs used by traditional clients.

After presenting the related work in Section 6.1, we explain the specificities of an opportunistic application client (intermittently connected to the Internet) that relies on multi-hop forwarding in Section 6.2. We present the security hazards that arise from these specificities and we show why solutions used to secure traditional (always connected) client applications are not sufficient in this case. Then, in Section 6.3 we propose a security architecture that takes

these issues into account and leverages some well-established security building blocks to offer a high level of security to the class of hybrid applications that we consider (*i.e.,* opportunistic applications with intermittent Internet connectivity).

6.1 Related Work

Several studies addressed the problems of security in opportunistic and vehicular networks (security issues in these two types of networks are similar with many respects). As a part of the European Haggle project [55, 56, 57] the authors present their work on a range of security mechanisms that target Haggle - a framework for autonomous opportunistic communication. However, their main focus are the security challenges raised by the class of content/context forwarding algorithms. Thus, they propose multiple solutions for correct forwarding operations over encrypted data. Another important difference between their work and the security solution we propose in this chapter is that they target a delay tolerant network in which nodes do not interact with legacy networks. Unlike them, we propose a security framework for the family of hybrid applications (that synchronize with the Internet) similar to the opportunistic Twitter application described in Chapter 5.

An important part of any security framework for a network that involves opportunistic (multi-hop) forwarding is node authentication. A number of papers address the problem of node authentication in an autonomous environment, without a trusted authority, where node security credentials are unknown or unobtainable. Solis et al. [59, 60, 61] propose a method for establishing an initial security context using casual information that links users to well-known entities. In [62], the system of invitations is used to expand the network in a trusted way. In [63], the authors propose a fully self-organized public-key management system that allows users to generate their public-private key pairs, to issue certificates, and to perform authentication regardless of the network partitions and without any centralized services. Their approach does not require any trusted authority. In [64], the authors present a method for key establishment over a radio link in peer-to-peer networks, based on the Diffie-Hellman key agreement protocol. They solve the problem of vulnerability to man-in-the-middle attack by leveraging on the ability of users to authenticate each other by visual and verbal contact. Finally, Asokan et al. [65] are closest to our work, as they investigate how security in DTNs can be bootstrapped from existing large-scale security infrastructure, like the cellular communication security infrastructure.

Unlike the proposals that address security of opportunistic networks, the major efforts to

secure vehicular communication approach the problem of node authentication with the assumption that there exists a central trusted authority. This assumption can be found in three major efforts to design vehicular security solutions that have been undertaken in industry and academia, namely: the NoW project [66], the IEEE 1609.2 working group [67], and the SeVeCom project [68]. They all rely on a Certification Authority (CA) and public key cryptography to protect vehicular communication, *i.e.,* to provide message authentication and integrity.

In addition to these major projects, a number of other notable studies outlined challenges [69], described attacks [70], and offered solutions [72, 73] in the domain of vehicular networks security. Some of them complement the public key operations with the use of symmetric key cryptography [72] or group signatures [73]. This is to a great extent motivated by the reduction of security footprint, discussed in Chapter 7.

6.2 Opportunistic Application Security Challenges

6.2.1 Security of Traditional Client Application

When publishing or fetching content, a traditional (always-connected) Twitter client application establishes an end-to-end session with Twitter servers. Since August 31st, 2010, third-party Twitter clients authenticate to Twitter servers using OAuth [108]. OAuth (Open Authorization) was introduced to avoid the exchange of login credentials (username and password) between different Twitter clients (*i.e.,* client applications implemented by different developers) and Twitter servers. It ensures that passwords are stored only on Twitter servers and are not accessible by third-party applications. The password is never stored locally by a client application and it is never exchanged between the client application and Twitter servers.

OAuth authorization works in the following way: A developer first has to register his client application with Twitter. As a result of this operation he is provided with two keys (a ConsumerKey and a ConsumerSecret) that uniquely identify the application. When a user first tries to login to Twitter using the developer's application, he will be redirected to the phone's web browser. He then enters his username and password, which are sent to Twitter via HTTPS. In exchange, the user receives two tokens - AccessToken and AccessTokenSecret. The client application stores the tokens in an Android shared preferences file, that cannot be accessed by any other application. The two application keys (ConsumerKey and ConsumerSecret) and the tokens (AccessToken and AccessTokenSecret) are used for every successive login to Twitter. Note that the exchange of password between the client and Twitter servers never occurs. Web

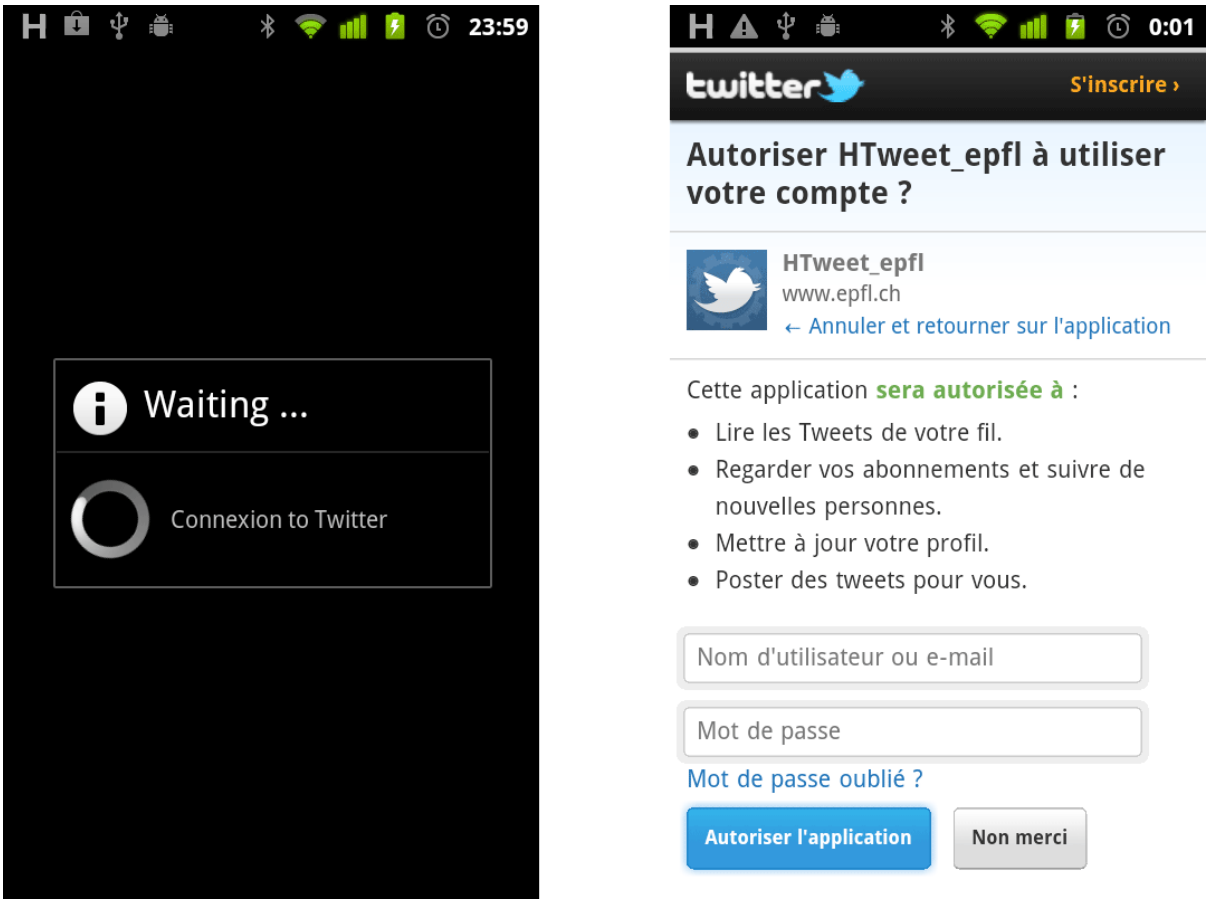


Figure 6.1: Initial login to Twitter: Obtaining AccessToken and AccessTokenSecret using OAuth authorization protocol. Note that username and password are sent using the mobile web browser and HTTPS. They are not accessible to the client HTweet application.

browser and HTTPS protocol are used instead. The login process with the Android implementation of our opportunistic Twitter client (HTweet), discussed in Section 5.2.4, is shown in Figure 6.1.

OAuth has de facto become an open standard for API access delegation. Today, it is a part of web services offered by major players on the web, such as Google, Facebook, Twitter, Microsoft, Yahoo, Netflix or LinkedIn. It allows users to share their private resources (*e.g.,* photos, videos, contact lists) stored on one site with another site without having to hand out their credentials, typically username and password. With OAuth, users hand out tokens instead of credentials to their data hosted by a given service provider. Each token grants access to a specific site (*e.g.,* a video editing site) for specific resources (*e.g.,* just videos from a specific album) and for a defined duration (*e.g.,* the next 2 hours). This allows users to grant a third party site access to their information stored with another service provider, without sharing their

access permissions or the full extent of their data.

Although this new security framework has several important advantages over the previously used security model (which required sending of username and password each time a client contacts a Twitter server), it is not applicable, as is, in the case of our opportunistic Twitter client. In the next subsection we discuss the reasons for this.

6.2.2 Specificities of Opportunistic Application Security

One of the major differences between a traditional (always connected) mobile client application and an opportunistic client application (with only occasional access to the Internet) is in the way data is forwarded. As discussed in the previous section, providing security for a client that can establish a connection with a server on the Internet at all times (by means of 3G, Wi-Fi, etc.) implies securing an end-to-end session. Data exchange is end-to-end, without participation of third-party entities in the middle. Encryption, authentication and protection from the man-in-the-middle attacks can be ensured using some standard solutions, such as Hypertext Transfer Protocol Secure (HTTPS). On the other hand, an intermittently connected opportunistic client often has to rely on other users (nodes) and epidemic (multi-hop) forwarding in order to exchange data with servers on the Internet.

Let us consider our opportunistic Twitter application described in Chapter 5 again. A user of this application has only occasional Internet connectivity, when direct communication with Twitter servers is possible. Most of the time, tweets created by the user are forwarded in opportunistic fashion, by other users, until they eventually reach a node with the Internet connectivity. Thus, the use of OAuth, such as described in Section 6.2.1, is impossible.

The problem is in the fact that most of the tweets get delivered to the Internet, not by their original creators, but by other users that participate in the process of forwarding. As Twitter's OAuth authorization requires creator's tokens (AccessToken and AccessTokenSecret) in order for a tweet to be published, these tokens would have to be attached to each opportunistically forwarded tweet. However, this is unacceptable, as allowing public access to users' private tokens creates serious security risks. Any user with access to private tokens of other users would be able to compromise the integrity of their tweets or to publish tweets on their behalf. In addition to this, such a user would be able to bypass any fairness scheme put in place and inject large number of junk messages in the opportunistic network on behalf of other users. This could lead to a full fledged Denial of Service (DoS) attack and reduced network performance.

6.3 Securing Applications with Intermittent Connectivity

6.3.1 Introducing a Trusted Location on the Internet

To address the aforementioned security and performance threats, a proxy server is added to our architecture (Figure 6.2). The place of the proxy server in the system and its non-security related roles are described in Section 5.2.5. As mentioned in the same section, it also has an important security role. This role stems from the following idea: Although an opportunistic user cannot share his private tokens with other users that participate in the forwarding process, he can make use of a trusted location on the Internet that is allowed to store his tokens and publish tweets on his behalf. Given that the Twitter API was designed with continuously connected users in mind, this trusted location (*i.e.,* our proxy server) serves as a buffer between the intermittently connected opportunistic clients and Twitter API. The proxy stores the tokens and keys of opportunistic users and publishes the arriving tweets on their behalf (see Figure 6.2).

By introducing the proxy server, we reduce the problem of secure publishing of opportunistic users' tweets to the problem of their uncompromised delivery to a trusted location on the Internet, by means of secure multi-hop forwarding. Security in this case means preserving message integrity, authenticity and possibly the ability to hide its content. These security issues, associated with the multi-hop context, have been addressed by the area of vehicular networks security [66, 68, 67]. Multi-hop vehicular communication is considered in the case of several transportation safety and efficiency applications, in particular congestion notification and environmental hazard notification applications [109, 71, 110]. As a result, suitable security frameworks were proposed. In the next section, we explain how we leverage on certain elements of these proposals [111], in order to adapt the PKI-based security solution to the needs of our application.

6.3.2 Securing Opportunistic Forwarding Using PKI

Like in the case of our opportunistic Twitter application, secure multi-hop communication in a vehicular network has to ensure the integrity of the forwarded messages and it has to provide a mechanism for user authentication. In other words, messages can be created only by legitimate users and their content should remain unchanged in the process of multi-hop (epidemic) forwarding. In order to secure these requirements, the architects of vehicular network security resort to some well established security building blocks, namely, Public Key Infrastructure (PKI). In the paragraphs that follow, we first discuss these building blocks in the

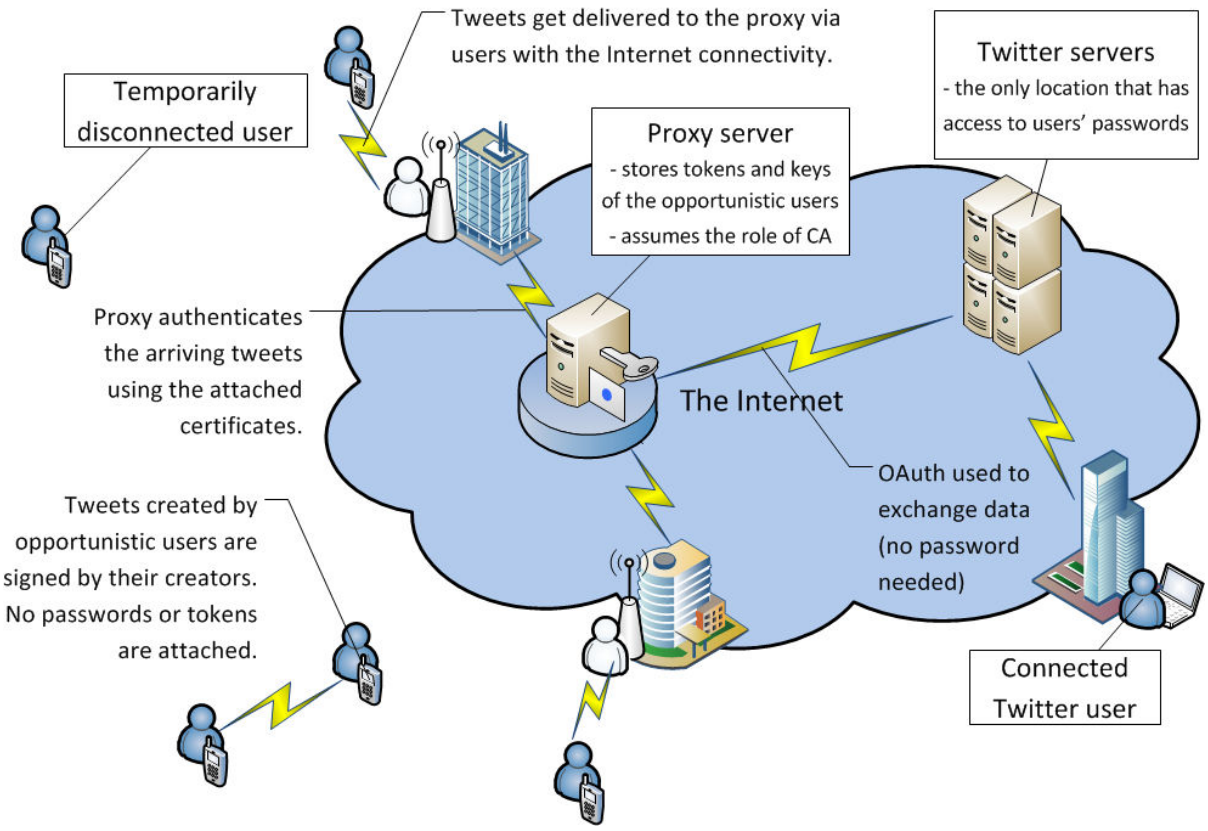


Figure 6.2: Security architecture for our opportunistic Twitter application.

context of vehicular networks security. We then explain their place in our opportunistic Twitter application.

As proposed in [66, 68, 67], user identities in a vehicular network are managed by a Certification Authority (CA). Each user in the network is assigned a set of private/public key pairs, with the latter being certified by a CA. Each legitimate user registered with the Certification Authority can participate in Vehicular Communication (VC). Basically, a user uses his private key to digitally sign a message he generates and the public key of the originator of a received message to validate the originator's signature. The state of the art in secure vehicular communication recommends the use of the fast Elliptic Curve Digital Signature Algorithm (EC-DSA) for message signing [68], [67]. The certificate of the message creator is attached to the message to enable validation of the message. The validation includes first the validation of the certificate (by validating the CA's signature), and then the validation of the creator's signature.

The described security architecture resolves the issue of authentication of the creator of a received message, as well as the problem of message integrity check, at each hop in the network. Let's see now how these different components of the public key infrastructure fit our

opportunistic Twitter architecture.

In the context of our opportunistic Twitter application, the role of the Certification Authority is naturally assumed by the proxy server. In our experiment, the proxy server is under our control, but in reality it can be controlled by Twitter itself or by a trusted third party that offers an interface to Twitter for users with opportunistic application clients. The proxy provides opportunistic Twitter users with public/private key pairs. The public key is certified by the proxy (the CA). The keys can be delivered to users following the procedure that is used in the case of OAuth token delivery. In other words, a mobile web browser and HTTPS can be utilized for the secure key delivery. In this process the users use their tokens (AccessToken and AccessTokenSecret) obtained from Twitter and the keys (ConsumerKey and ConsumerSecret) provided by the application developer (that he also obtained from Twitter) to identify themselves to the proxy.

Note that there is no need for users to share their Twitter passwords with the proxy server, as the proxy can exchange data with Twitter using only OAuth keys and tokens. In order to enable users to authenticate the proxy, the proxy's root certificate can be delivered to the application client developer prior to the application launch, following the method used for the OAuth key delivery (ConsumerKey and ConsumerSecret). Just like the OAuth keys and tokens, a user's private key can be stored in a shared preferences file, access to which is restricted to a single application.

6.3.3 Possible Implications on Network Performance

At the time of our experiment (the end of 2009) OAuth authorization was still not deployed by Twitter. Users' usernames and passwords were used for authentication with Twitter, instead of OAuth keys and tokens. From the aspect of the proposed security architecture, this meant that the proxy server stored users' passwords instead of OAuth keys and tokens. Proxy had the same role, but it used a user's password to exchange data with Twitter on his behalf. Nevertheless, the PKI part of the proposed security framework is not affected by this difference in Twitter API.

However, during the experiment, the focus was on the performance evaluation of a live opportunistic application and comparison with the performance obtained from the contact based simulations. Thus, given the limited processing power of our Windows Mobile phones and the performance footprint introduced by asymmetric key cryptography [112], the PKI support offered by Haggle [58] was switched off, for the whole duration of the experiment. Nonetheless,

the proxy was the only entity with the access to users' credentials, which were never included in the exchanged messages. This was, of course, not sufficient to fully meet the discussed security requirements, as the messages were not signed or validated. This means that tweets could have been modified in the process of opportunistic multi-hop forwarding. Also, the application users could have been impersonated with some application client tampering. In spite of this (perhaps due to the fact that the experiment participants did not have access to the application source code), no such cases were detected.

The results of our comparison between the experiment and contact-based simulations presented in Chapter 5 show that even without the additional security load on performance (added by the use of asymmetric cryptography), data exchange opportunities that appear during the opportunistic contacts are often missed. Thus, an interesting question arises: Should we expect further performance deterioration, once the security footprint is added?

Network simulators that model all network layers and go beyond simple contact-based simulations [113], [114], in combination with the plugins developed for vehicular networks [115], offer a good environment to study this problem. Given the similarity between a system with vehicular nodes that participate in multi-hop forwarding and a setup with an opportunistic application (also based on multi-hop forwarding), in Chapter 7, we try to model the problem in a way that covers both scenarios. Using the results of previous measurements and extensive simulation, we show that the problem of reduced network performance (*i.e.,* reduced relaying capacity), due to the additional security-related processing load, effectively exists. For this reason, we design and evaluate an adaptive scheme that complements the proposed security framework and protects the relaying capacity in the network.

6.4 Conclusion

Hybrid opportunistic application clients that occasionally access resources on the Internet (that are designed primarily for the access by traditional always-connected clients), impose highly specific security requirements. On one hand, as they rely on epidemic forwarding, authenticity and integrity of the content, which is often forwarded over multiple hops, has to be ensured. On the other hand, interfacing with the web services (resources) in question and adherence to their security APIs has to be secured. For these reasons, neither the proposals for securing the autonomous opportunistic communication, nor the existing security solutions for the always-connected application clients can be used, as are, to secure this class of hybrid applications.

Thus, we propose a security framework for opportunistic application clients with intermittent access to the Internet, which is based on existing security building blocks and which addresses both sets of requirements. To manage the communication with the security API of the web service in question, we introduce a trusted entity on the Internet (our proxy server). This trusted location is designed in a way that allows asynchronous access to intermittently connected opportunistic clients. It also serves as a trusted certification authority, enabling secure exchanges among opportunistic clients.

Chapter 7

An Adaptive Method for the Optimization of Security Performance

Recent benchmarks indicate that the use of public key cryptography results in non-negligible verification times on a variety of platforms [112]. Complex cryptographic operations, such as signature verification, introduce non-negligible processing delays. In this chapter, we focus on multi-hop communication in opportunistic and vehicular environments and we show that the increase in message processing time in mobile nodes degrades network performance, by decreasing the number of messages that reach destinations. Nevertheless, ignoring security can lead to DoS attacks and an even more severe decrease in network performance [55]. As a solution to this problem, we design Adaptive Message Authentication (AMA), a lightweight filtering scheme that addresses the problem of security footprint (introduced by the security framework proposed in Chapter 6) and protects the threatened relaying capacity. Although based on local observations and without any additional communication channel between the nodes, our scheme achieves global improvement of network performance. We perform extensive simulations and show that the scheme resists DoS attacks even against a substantial number of adversaries in the network.

In Chapter 6, we have discussed the similarities in security requirements, in the cases of opportunistic applications that require multi-hop forwarding and vehicular applications that rely on inter-vehicle communication (IVC). These similarities led us to import certain elements of the PKI-based security solution, proposed in the vehicular context, into our security framework for intermittently connected opportunistic applications. Both security architectures advocate an authentication and integrity check of each relayed message as a necessary condition for secure

multi-hop communication. In this chapter we revisit this requirement, in order to explore if it is possible to reduce the security footprint on performance and optimize the relaying capacity in multi-hop applications, without affecting the level of security.

The rest of this chapter is organized as follows. After presenting the related work in Section 7.1, we explain the system assumptions (considered applications, forwarding, security assumptions) and the adversary model in Section 7.2. We discuss the impact of security footprint on performance in Section 7.3. In Section 7.4, we introduce our scheme for security footprint reduction, which we dub Adaptive Message Authentication (AMA). Finally, in Section 7.5, we show how our scheme can be configured to protect the optimal relaying capacity and to account for the possible security threats. We conclude the chapter in Section 7.6.

7.1 Related Work

On one hand, the toll of operations associated with public key security is well illustrated in [112], where durations of different cryptographic operations on a variety of platforms are measured. On the other hand, the simulation study in [55] shows the effects of denial of service attacks on epidemic forwarding protocols, when no security solutions are deployed.

The proposals for security footprint reduction in opportunistic or vehicular networks are typically coupled with the proposals that offer security solutions for these networks. After describing a method for establishing an initial security context in an autonomous opportunistic network, using social contact information, Solis et al. [59, 60, 61] relax the authentication requirements in order to reduce security overhead. In [72], symmetric key cryptography complements the public key operations, as a part of the effort to reduce security footprint in vehicular networks. In [73], the authors use group signatures and simple, context-agnostic overhead reduction schemes, to complement public key cryptography. Investigating these variants, which result in somewhat different processing loads, (in the context of the scheme we propose in this chapter) would be an interesting point for future work. At this point our scheme is evaluated only in the case of security solutions that are based exclusively on public key cryptography, *i.e.,* the framework proposed in Chapter 6. However, we note that the scheme is oblivious to the exact use of certificates and public keys. As a result, AMA can remain fully operational and effective even if privacy enhancing algorithms with multiple certified public keys (pseudonyms) are implemented.

In [74, 75], the authors propose context-specific strategies for security footprint reduction. The investigation of the vehicular communications security footprint and its effects on sys-

tem/application performance is further extended in [76], where both safety and efficiency applications and additional security mechanisms are considered. These works are complementary to the security footprint reduction scheme that we propose and their joint investigation with AMA (our scheme) would be another interesting point for future work.

Regarding the geographic routing protocols that are used in this chapter (and that represent a special case of epidemic forwarding), a good survey can be found in [116]. The security aspects of these protocols are explored in [117] and [118].

7.2 System and Adversary Model

7.2.1 Considered Applications

We consider a system with mobile nodes that represent either: (i) a population of mobile users with an opportunistic smartphone application that relies on multi-hop forwarding or (ii) a population of vehicles (equipped with a variant of IEEE 802.11, on-board sensors, and a computing platform) that participate in inter-vehicle communication.

The considered smartphone application is similar to our opportunistic Twitter application, discussed in Chapters 5 and 6. It allows mobile users to create their messages (tweets for example) and to have them forwarded in the direction of recipients or data sinks (*i.e.,* other mobile users with an Internet connection). The only difference with respect to our opportunistic Twitter application is that here it is assumed that each message contains coarse destination regions or target areas where the recipients (or data sinks) can be found. This assumption allows us to perform our analysis using more comprehensive routing protocols, suitable for this kind of environment, which introduce less overhead than simple epidemic forwarding.

Regarding the second scenario, multi-hop vehicular communication is often considered in the context of congestion notification [119, 120] and environmental hazard notification applications [110, 109, 71]. These applications exploit the inter-vehicle communication, which will be based on a variant of the currently widely used IEEE 802.11 protocol. The IEEE 1609.x protocol suite, also known as the WAVE technology [67] developed for the U.S. Department of Transportation (DOT), is already in the stage of a trial-use standard. Vehicles will communicate with other vehicles within range, but, equally important, they will cooperate in forwarding messages of their neighbors, other vehicles or road-side units (RSUs), across multiple hops.

7.2.2 Position-Based Routing

The type of routing that is shown to be efficient in the case of multi-hop forwarding with known destination regions is position-based routing [116]. These routing algorithms, often termed GeoCast protocols, are currently under consideration towards standardization [71]. The messages forwarded using GeoCast routing contain the destination or target geographic areas. Position-based routing protocols appear as a natural choice for multi-hop communication, as nodes are expected to be aware of their own location (through GPS, or other localization techniques with respect to terrestrial infrastructure). In such a setting, the highly volatile topology makes these protocols more efficient than the other mobile ad hoc routing protocols [121].

In the case of vehicular communication (VC), multi-hop forwarding serves, in general, less time-critical applications. In addition to the multi-hop communication, VC-enabled applications entail one-hop high-rate safety messaging (beaconing), with messages bearing information on the location, speed, acceleration and heading of the sender. Beacons are transmitted at a rate of 3 to 10 beacons per second. Safety beaconing rates are specified by standards, and the processing of the beacons is obligatory. It allows a node to maintain a fine-grained knowledge of the motion dynamics of other vehicles in its vicinity.

In order to account for these differences between the observed opportunistic applications (that rely exclusively on multi-hop forwarding) and vehicular applications (that also involve one-hop beaconing), we consider two essentially different position-based algorithms.

The first algorithm, Cached Greedy GeoCast (CGGC) [122] belongs to the group of beacon-based unicast routing algorithms. CGGC relies on beacons to discover the position of neighboring nodes (within the communication range), and then forwards messages in the geographic direction of the destination, picking the node whose coordinates are the closest to the destination. If a local optimum is reached, the message is added to the local cache, where it is kept until a suitable next hop is found. As it includes the obligatory beaconing, this algorithm is well suited for VC-enabled applications.

The second algorithm we consider is Contention-Based Forwarding (CBF) with the basic suppression scheme based on timers [123]. Unlike CGGC, CBF is based on broadcast and performs greedy forwarding without the help of beacons and neighbors' tables. It leaves the next hop selection and the forwarding decision to the neighbors in the transmission range. Thus, it fits in the context of the mobile opportunistic applications discussed in Chapters 5 and 6.

7.2.3 Goodput as the Performance Metric

Here we define the performance metric that is used in the rest of this chapter. Let us define L as the set of legitimate users running applications enabled by multi-hop communication and N_i as the number of legitimate messages received by destination i over the time period of interest. We consider here multi-hop transmissions originating at each node at a constant rate of r_L messages per second. Then, in the presence of any communication impairments and networking faults and delays, we define the *goodput* γ_L as:

$$\gamma_L = \frac{1}{|L|} \sum_{i \in L} \frac{N_i}{\text{total time}} \tag{7.1}$$

Goodput as a metric is more meaningful than delivery ratios in the context of the DoS attacks that we consider in Section 7.2.4. In other words, it is often more meaningful to use goodput as a metric in the scenarios that involve adversaries who inject forged messages in the network. Unlike delivery ratios, goodput measures only the arrivals of legitimate messages at their destinations. Thus, it can better capture the drop in performance caused by the injection of forged messages.

7.2.4 Security Assumptions and the Adversary Model

We consider the security framework based on public key cryptography, presented in Chapter 6. To summarize, user identities in such a setting are managed by a certification authority (CA). Each user (node) in the network is assigned a set of private/public key pairs, with the latter being certified by the CA. Users use their private keys to digitally sign the messages they generate and the public key of the creator of a received message to validate the creator's signature. To facilitate the process of message validation, the certificate of the message creator is attached to each message. The validation includes first the validation of the certificate (by verifying the CA's signature), and then the verification of the creator's signature.

We focus on the external adversaries, that is, adversarial users that do not have in their possession system credentials (certificates issued by the CA). Each such adversarial node can fabricate and inject messages, but cannot sign on behalf of a legitimate node. The direct goal of adversaries is to reduce the goodput γ_L of legitimate nodes by injecting forged messages and making legitimate nodes waste their processing time on forged message verification. All adversarial nodes inject forged messages at a rate r_A messages per second. We assume that the adversaries are aware of their number in relation to the number of legitimate nodes in the

network and we define a as the percentage of adversaries in the network. Knowing a, the adversaries choose their sending rate r_A in order to minimize γ_L.

We emphasize that none of the forged messages injected by an adversarial node can be perceived as valid if it is checked by a legitimate node. Nonetheless, the overhead imposed by the need to validate those messages is exactly what can lead to a DoS attack. For such an attack, a smartphone or a laptop can be used, and no tampering with hardware that stores the vehicle's cryptographic keys [68, 70] and no cryptanalytic attack are necessary. By preventing even a small fraction of legitimate traffic from reaching its destination, the adversary can prevent reception of messages in an area within the necessary deadlines: For example, consider road condition information that cannot be validated in the targeted geographical area, resulting in traffic jams; or, consider increased loads from fabricated traffic that prevent a node from performing safety related operations.

7.3 The Impact of Security Footprint on Performance

7.3.1 Delay Introduced by Message Validation

The message validation in a node is defined in Section 7.2.4. In the rest of this chapter, we refer to the process of message validation as "message checking." We denote the processing delay needed for a message to be checked by t_C. The value of this delay depends on the selection and implementation of the algorithm used for message signing. The algorithm that is typically considered, due to the fact that it introduces lower delay than the more famous RSA, is the Elliptic Curve Digital Signature Algorithm (EC-DSA).

The processing delay also depends on the processing power of the device that performs the validation. For platforms similar to those currently considered for the proof-of-concept implementations of VC systems, characteristic delays are provided in [73, 118]. For platforms based on PowerPC microprocessor (*e.g.,* , DENSO platform [124]) the delays can be found in the eCrypt project benchmarks [112]. For example, we can see that on 533MHz CPU Power PC platform, signature verification for EC-DSA with 192-bit curve (nist-p-192), requires 9 ms on average. The benchmarks for the widely used Crypto++ library [125] show that even on Intel Core 2 1.83 GHz processor (under 32-bit Windows Vista) signature verification for EC-DSA with 256-bit curve takes 8.53ms, while in the case of the 233-bit curve it takes 12.8ms. Lastly, the IEEE 1609.2 efforts currently consider, for proof-of-concept purposes, hardware-accelerated signature verification at several milliseconds. Even for the most powerful of these

platforms, t_C would be 7.2ms [73]; if we consider this value in a rather favorable environment, with 20 neighbors beaconing at the lowest possible rate of 3 beacons/second, by multiplying the three numbers, we obtain as a result that 43.2% of the CPU time would be devoted to message checking.

7.3.2 Security Footprint of Multi-Hop Forwarding

The use of multi-hop forwarding by opportunistic and vehicular applications implies that messages can traverse several intermediate hops before reaching the destination. Given the message validation delays described in Section 7.3.1, a question arises: What strategy should an intermediate node adopt with regard to checking the messages that are only relayed by that node? It is not a priori clear whether a message that requires relaying should be checked by an intermediate node or just resent without any prior validation.

We define "check-all" and "check-nothing" as two extreme approaches that can be applied to relayed messages. "Check-all" is the default strategy in the existing proposals [111] and it assumes checking of each relayed message, whereas "check-nothing" assumes that none of the relayed messages are checked. Both approaches perform well under certain circumstances, but underperform in other cases.

The "check-all" strategy guarantees the fewest forged messages in the network, as it contains them locally and prevents their propagation. Given unlimited processing power in each node (implying a negligible checking time t_C), checking each relayed message would be the best strategy. In this case, less time would be spent forwarding the forged messages and the limited wireless capacity would be used only for forwarding the valid messages.

The "check-nothing" strategy promises good results with few adversaries in the network, or with few injected forged messages in the network. This is shown in Figure 7.1, which contains a snapshot of our performance evaluation results provided in Section 7.5.3. In this case, not checking any relayed traffic guarantees that no time is wasted on signature verifications in intermediate nodes and the goodput γ_L of legitimate users is improved. However, an increase in the number of adversaries in the network quickly makes this strategy inferior to "check-all".

In conclusion, "check-all" is not the best approach for networks with few adversaries and "check-nothing" gets worse as the number of adversaries increases. We want a scheme that performs well in both cases. It should contain the forged messages locally (as "check-all") in the presence of adversaries and behave as "check-nothing" with no adversaries around.

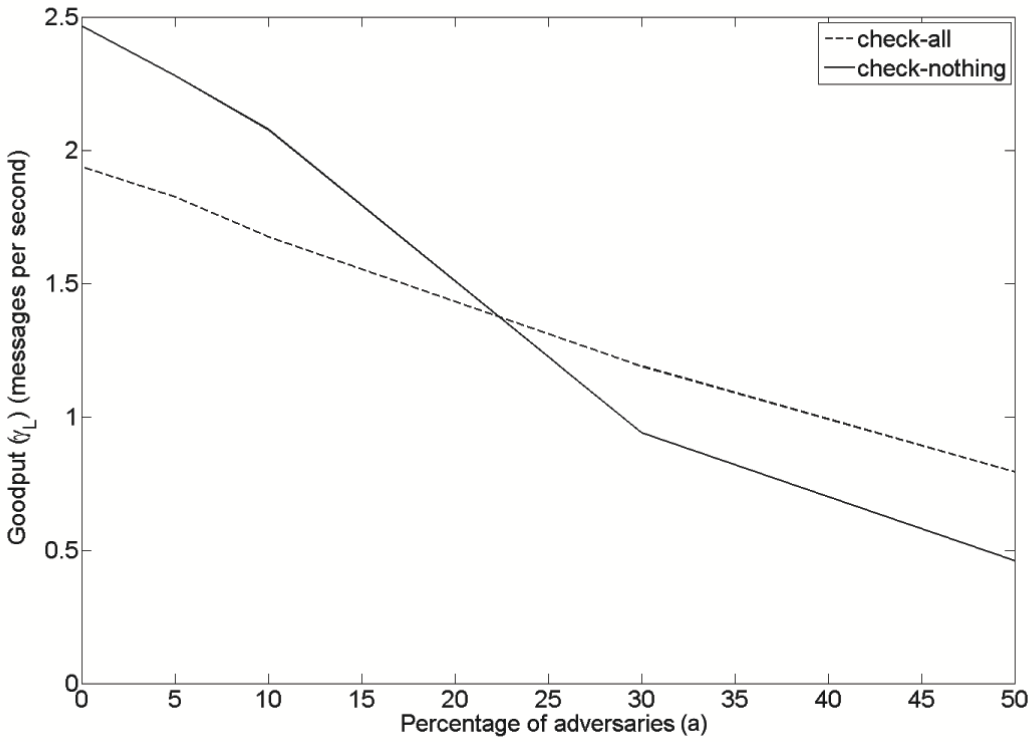


Figure 7.1: The performance of "check-all" vs. "check-nothing" algorithm obtained for CBF geocast routing algorithm.

7.4 AMA-Adaptive Message Authentication

As explained in Section 7.2.4, the adversaries can try to degrade the performance of a system that involves multi-hop forwarding, by creating forged messages. In principle, the danger can be twofold: (i) the use of forged messages at destinations and (ii) the reduction in goodput caused by flooding of the network with forged messages. The first risk is not really a concern, as all solutions require nodes to check each message at destination, prior to its use. The impact of the second threat depends critically on the ability of the security solution to prevent the spatial propagation of forged messages.

The existing solution ("check-all") is very restrictive with respect to this issue. Messages are checked at each hop, which prevents forged messages from propagating in the network. However, in the absence of adversaries, this results in many redundant checks that consume processing time and reduce performance. Indeed, in Figure 7.1 (where "check-all" and "check-nothing" strategies are simulated for different fractions of adversaries in the network), we

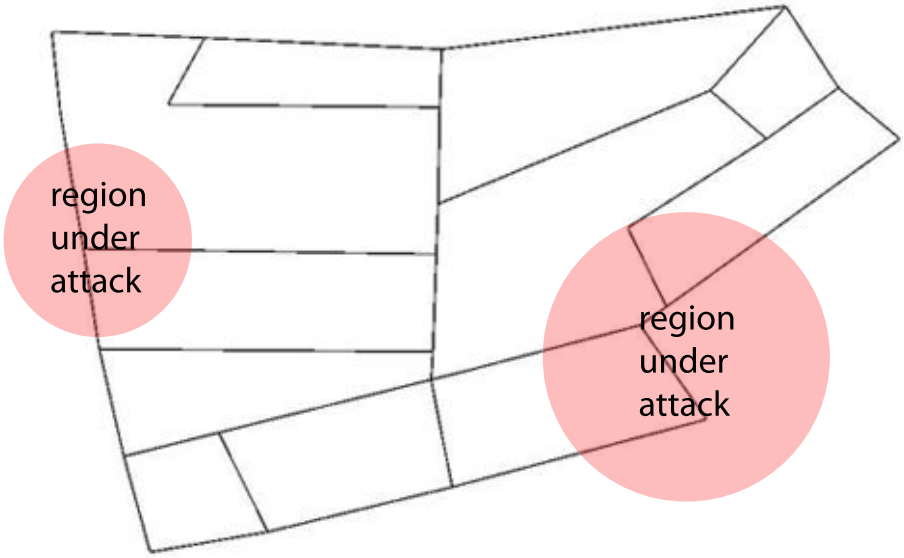


Figure 7.2: A few regions in the city are under attack (shaded areas) and nodes in these regions actively defend. Nodes in the rest of the city, where the presence of the adversaries is not felt, can relax their security. We use this road map in all of our simulations.

observe that "check-nothing" performs significantly better in terms of goodput without adversaries in the network. Thus, our goal is to provide a solution that takes the best of both strategies. The aim is to make nodes perform only the necessary number of cryptographic operations while skipping the redundant message checks and improving the overall performance of the network.

The basic idea that we exploit is based on the observation that the adversaries are limited in scope and that they cannot keep the whole network under attack at all times [126, 70]. Figure 7.2 illustrates this; it contains the street grid used in our simulations; the regions under attack at time t are marked as shaded areas. So, if the security conditions in different parts of the network significantly differ, why would the nodes in these areas behave in the same manner? In other words, we argue that in this case nodes should take reactive, rather than proactive approach to security. A node should respond to a threat only when it is affected by an attack and it should reduce its security-related activities when the threat is not present (knowing that these activities consume resources and reduce performance).

The solution we propose is an adaptive scheme that is shown in Figure 7.3. We call it AMA (Adaptive Message Authentication). When no threat is present, our scheme relaxes security and avoids the unnecessary processing load. Nevertheless, it maintains the ability to discover a threat and it becomes very conservative as soon as it is detected. AMA has two modes of

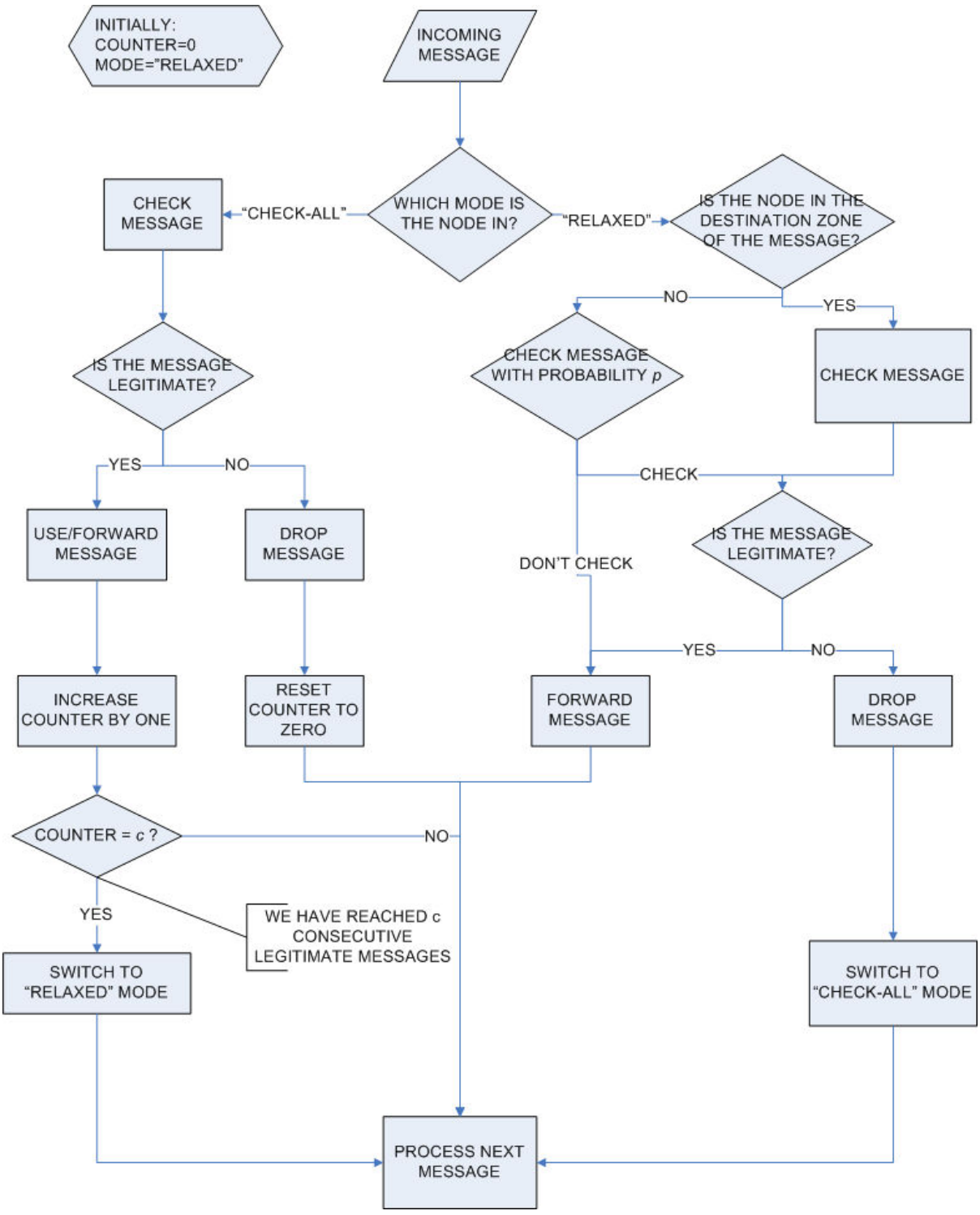


Figure 7.3: AMA - the scheme for adaptive authentication and integrity checking of messages exchanged between vehicles. Briefly, an AMA node can be in one of two modes: "check-all" and "relaxed." A node starts in "relaxed" mode. In this mode, a node checks with probability 1 the messages destined for itself, but only with probability p the messages destined for other nodes. If it detects a forgery, the node switches to the "check-all" mode. In the "check-all" mode, a node checks all messages with probability 1, and switches to "relaxed" mode only if c consecutive legitimate messages are received.

operation. We call them "check-all" and "relaxed".

The "relaxed" mode allows nodes to spend less processing power on defensive measures. All the legitimate nodes are initially in the "relaxed" mode. It is this mode that is expected to improve the performance of the scheme, as only a fraction of received messages are checked by a node in the "relaxed mode". Nodes distinguish between the messages that have the current location of the node as the destination zone and those that only have to be relayed to others. Each message in the first group is checked with probability 1 and each message in the second group with probability p. If they happen to check a forged message, the forgery is always detected and it forces the node to switch its mode of operation to "check-all".

The "check-all" mode is conservative and it mandates checking each received message. A legitimate node is expected to be in this mode when there are adversarial nodes nearby. A node stays in the "check-all" mode until it receives c consecutive legitimate messages. Then, it switches back to the "relaxed" mode.

The rationale is that if a node senses that the current "temperature" of the neighborhood is low (no adversaries in the neighborhood), a node can relay most of the messages without prior authentication and integrity check, while checking only a small fraction of these messages in order to ensure a timely detection of security threats. While the selected messages are being checked, the other messages that need to be relayed do not have to wait before being forwarded.

It is possible, of course, to use a different function for the checking rate increase, not just a step function. We show that even this simple scheme guarantees significant performance gains, for an appropriate choice of the parameters p and c, under very realistic assumptions (the scheme and both parameters p and c are known to the adversary).

7.5 Protecting the Optimal Relaying Capacity

7.5.1 Simulation Setup

To evaluate our scheme, we compare the goodput γ_L (as defined in Section 7.2.3) achieved when AMA, "check-all," and "check-nothing" strategies are used. The strategies are tested on top of the two geocast algorithms described in Section 7.2.2 (Cached Greedy GeoCast (CGGC) [122] and Contention Based Forwarding (CBF) with the basic suppression scheme based on timers [123]). The broadcast based CBF algorithm covers the opportunistic application scenario, while the CGGC algorithm that includes obligatory one-hop beacons covers the vehicular scenario.

In order to make the simulations as realistic as possible, we use a traffic simulator to simulate user mobility and a network simulator to capture the properties of the wireless environment. For generating mobility traces, based on road network topologies obtained from the real maps, we use the SUMO traffic simulator (v 0.9.8) [127] with the TraNS extension [128, 129]. User speeds are limited by the legal speed limits in the part of the lower Manhattan that is used as the road topology. This topology is shown in Figure 7.2. It covers about 6 sq. km and it is populated by 600 nodes in our simulations.

The mobility traces generated for this road topology are passed to the SWANS network simulator. SWANS is the ad hoc network simulator developed on top of the JiST discrete-event simulator [114]. Unlike the contact-based simulations discussed in Chapter 5, SWANS allows us to model the physical layer and signal propagation. We implemented the full TCP/IP stack in each node, with the exception of the transport layer. At the physical layer we use the two-ray pathloss model, which incorporates ground reflection. At the link layer, 802.11b is used. The SWANS implementation of 802.11b includes the complete DCF function, with retransmission, NAV and backoff functionality.

For the "check-all" and "check-nothing" strategies, we run 20 simulations with every combination of the percentage of adversaries $a \in \{0, 5, 10, 30, 50\}$ and the adversaries' sending rates $r_A \in \{0, 1, 2, 5, 10\}$ messages per second. For AMA we run 20 simulations for every combination of the percentage of adversaries $a\%$, the adversaries' sending rates r_A, and AMA parameters $p \in \{0.05, 0.1, 0.2, 0.3, 0.5\}$ and $c \in \{20, 40, 60, 80, 100\}$. For each run, we randomly select $a\%$ adversaries out of the total set of nodes in the network. For each generated message, a destination region is selected at random. The legitimate node sending rate r_L is 1 message per second. The size of each geocast message is 300 bytes. The checking time t_c is $10ms$. Each simulation lasts for 500 seconds of simulation time.

The data traffic starts 5 seconds after the start of the simulation, giving the beacon-based protocol enough time to exchange neighbor information and it ends 15 seconds before the end of simulation, leaving enough time to nodes to deliver remaining messages that are waiting in queues, before the simulation is terminated. Greedy forwarding with beacons is simulated with the ability to re-route messages in case the link layer signals to routing that the next hop is not reachable any more.

Beacon verification is extremely important for vehicular security. Thus, as explained in Section 7.2.2, we assume that nodes check all received beacons in the simulated beaconing based routing algorithm (CGGC). Since beacons make a large percentage of the total traffic in all the beaconing-based routing algorithms, we want to simulate the load that beaconing traffic

puts on the processor in a realistic way. For this reason, the CGGC beaconing rate that we use in our simulations is 1 beacon/300ms, *i.e.,* , the value likely to become a part of the standard.

7.5.2 Min-Max Parameter Selection

Having calculated, through the simulations, the goodput γ_L achieved under AMA for all combinations of the parameters (p, c, a, r_A), we now show how to preselect the parameters p and c to maximize it. In making this selection, we have to keep in mind two things:

- The percentage a of adversaries is fixed, but may or may not be a priori known. Below, we distinguish two cases according to whether it is known or not.
- The adversaries will learn the selected values of p and c, and choose their sending rate r_A to minimize the goodput.

In the first case, which we call the *pessimistic* case, we do not know the percentage of adversaries a. So, we select the parameters p and c that maximize the resulting goodput γ_L against the worst case combination of the percentage of adversaries a and their sending rate r_A.

$$(p^*, c^*) = \text{argmax}_{(p,c)} \min_{(a,r_A)} \gamma_L (p, c, a, r_A) \tag{7.2}$$

To visualize this selection, consider Table 7.1, where each entry is equal to the goodput achieved with the corresponding row-column combination of parameters.

	(a^1, r_A^1)	(a^1, r_A^2)	$\dots$	(a^1, r_A^k)	(a^2, r_A^1)	$\dots$
(p^1, c^1)	γ_L^{1111}	γ_L^{1112}	$\dots$	γ_L^{111k}	γ_L^{1121}	$\dots$
(p^1, c^2)	γ_L^{1211}	γ_L^{1212}	$\dots$	γ_L^{121k}	γ_L^{1221}	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
(p^1, c^m)	γ_L^{1m11}	γ_L^{1m12}	$\dots$	γ_L^{1m1k}	γ_L^{1m21}	$\dots$
(p^2, c^1)	γ_L^{2111}	γ_L^{2112}	$\dots$	γ_L^{211k}	γ_L^{2121}	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 7.1: Maxmin selection of AMA parameters: We select the parameters p and c that maximize the resulting goodput γ_L against the worst case combination of the percentage of adversaries a and their sending rate r_A.

First, Eq. (7.2) selects the minimum element in each row. The minimum element in a row is the goodput value that will be achieved if we select the (p, c) pair of that row, and the adversaries' percentage happens to be the one in the minimizing column. If the adversaries' percentage is the right one (worst case scenario for the legitimate users), they can choose their sending rate to achieve the worst case goodput.

Then, among the minimum elements found, Eq. (7.2) chooses the largest one by selecting the appropriate (p, c) pair. This way, we can guarantee the adversaries would not achieve a lower goodput, even if they could change their percentage.

In the second case, which we call the *optimistic* case, we know a but not r_A. So, for the given value of a, we choose p and c that maximize γ_L against the worst case reply r_A.

$$(p^*, c^*)(a) = \text{argmax}_{(p,c)} \min_{r_A} \gamma_L \left(p, c, a, r_A\right) \tag{7.3}$$

Referring to the previous explanatory table, Eq. (7.3) now does the same minimization-maximization as Eq. (7.2), but operates on the columns corresponding to the known value of a. In either case, the adversaries choose their sending rate r_A to minimize γ_L, given the legitimate users' choice of p^* and c^*:

$$r_A^*(a) = \text{argmin}_{r_A} \gamma_L \left(p^*, c^*, a, r_A\right) \tag{7.4}$$

$$r_A^*(a) = \text{argmin}_{r_A} \gamma_L \left(p^*(a), c^*(a), a, r_A\right) \tag{7.5}$$

Note that the adversaries do not optimize over the percentage a, as they cannot change it.

7.5.3 Performance Evaluation

We find that AMA outperforms "check-all" and "check-nothing" strategies for all the considered values of a, regardless of the routing algorithm. The goodput obtained with the CBF routing algorithm is shown in Figure 7.4. The goodput obtained in the case of the CGGC routing algorithm is shown in Figure 7.5.

The curves shown in the figures are the mean values of 20 simulations and the error bars extend a standard deviation above and below the mean values. Note that the knowledge of the percentage of adversaries a (which is not easy to obtain) guarantees only a slight performance improvement (the pessimistic scheme performs almost as good as the optimistic).

As we can see from the figures, under pessimistic AMA, which assumes no knowledge about the number of adversaries or their sending rate, the goodput of legitimate nodes γ_L improves up to 30% for CBF and up to 33% for CGGC routing.

In the case of CBF, the performance gain is due to the reduction in the number of messages checked in the intermediate nodes. Our simulation data shows that the number of checked geocast messages drops up to 46% percent in this case.

Apart from the drop in the number of cryptographic operations, the performance of CGGC

routing algorithm is affected by the beacon processing load. Geocast messages now share the CPU time with beacons. Checking or not checking a geocast message or a group of messages can make the difference between an immediate check of another arriving geocast message and its prolonged stay in the intermediate node due to the CPU busy period introduced by beacons. The same applies to forged messages, as their increased number in the incoming queue can make valid messages wait for a period of time before being checked and relayed. This is the main reason why the performance of "check-nothing" strategy drops with the increase in the number of adversaries in the network.

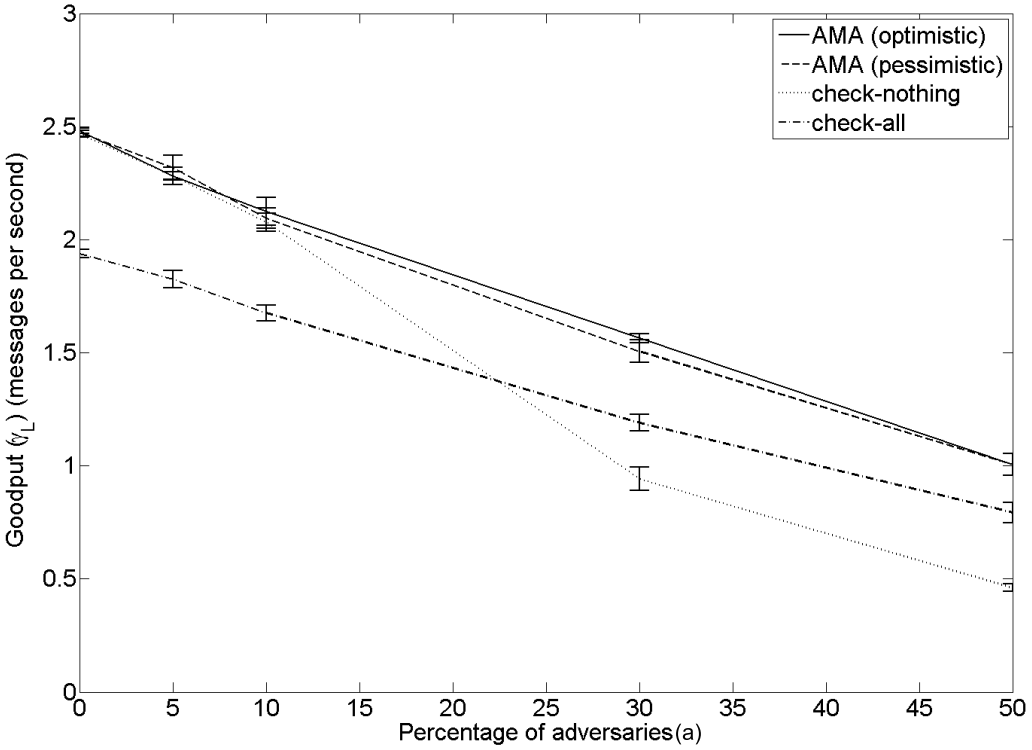


Figure 7.4: Goodput γ_L obtained under (pessimistic and optimistic) AMA, "check-all", and "check-nothing" strategies for the CBF routing algorithm.

The introduction of other CPU tasks (not related to forwarding) would make this effect even more visible. Since smartphones and vehicular security units have to share the CPU time with other tasks, the moment when an incoming message receives its share of CPU becomes extremely important. A single forged message or an unnecessary check of a legitimate message can make the arriving geocast message wait for an additional few hundred milliseconds due to the CPU multitasking.

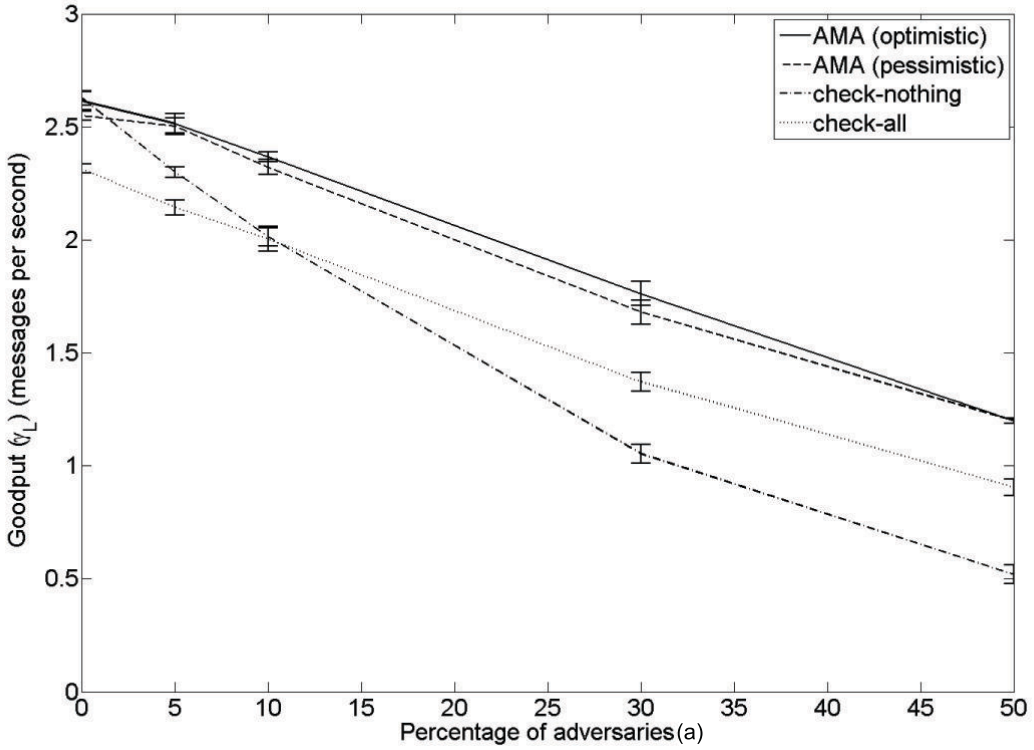


Figure 7.5: Goodput γ_L obtained under (pessimistic and optimistic) AMA, "check-all", and "check-nothing" strategies for the CGGC routing algorithm.

In the case of pessimistic AMA, the optimal values of p and c obtained using the described maxmin approach are $p = 0.2$ and $c = 40$ for both CGGC and CBF routing algorithms. In the optimistic case, the obtained values for p and c, for both considered algorithms, are shown in Table 7.2.

Percentage of adversaries	CBF		CGGC	
	p	c	p	c
$a = 0$	0.05	60	0.2	80
$a = 5$	0.05	20	0.05	80
$a = 10$	0.2	100	0.3	80
$a = 30$	0.3	100	0.2	40
$a = 50$	0.2	40	0.2	40

Table 7.2: The optimal parameters p and c for *optimistic* AMA.

If the adversaries have less knowledge than we assumed (*i.e.,* if they do not know the p^* and c^*), they may choose a sending rate other than the computed optimal r_A^*. In Figures 7.6 (for

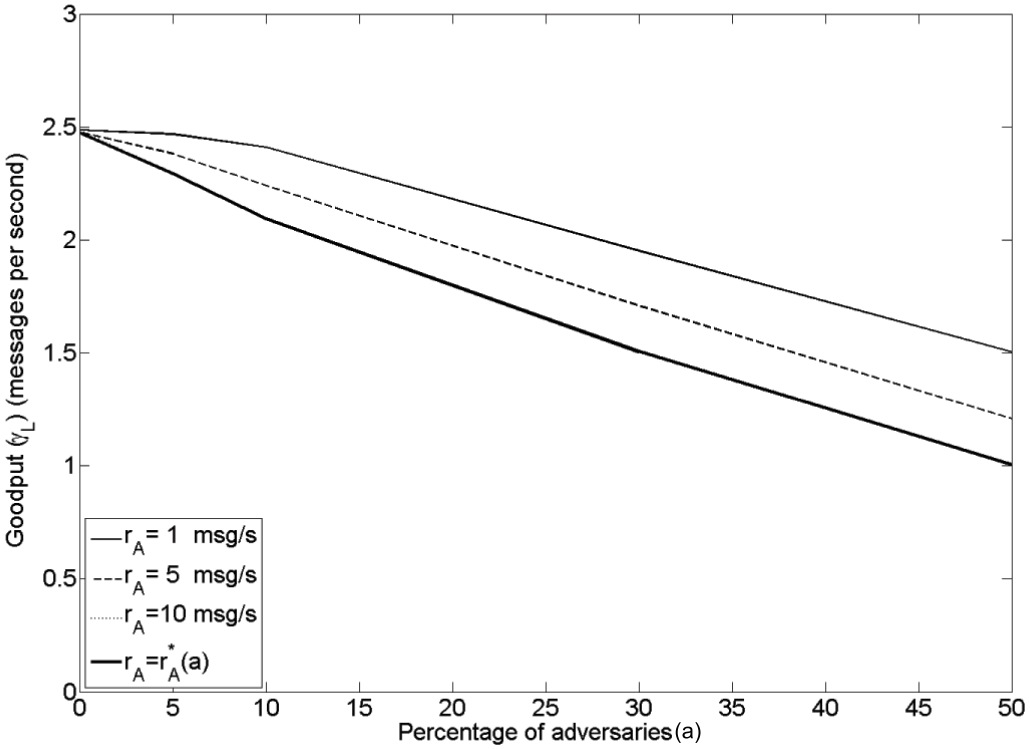


Figure 7.6: Goodput γ_L obtained for various adversarial sending rates r_A under pessimistic AMA for CBF routing algorithm. The thick line corresponds to the optimal sending rate r_A^*.

CBF) and 7.7 (for CGGC) we plot the resulting γ_L-a curves for different sending rates r_A for the pessimistic choice of p^* and c^*. We see that a suboptimal selection of the sending rate by the adversaries results in improved performance for our scheme. The thick line in the figures corresponds to the optimal (for the adversaries) sending rate r_A^* and represents the worst case scenario (*i.e.,* the lower bound for the goodput).

7.6 Conclusion

The security framework based on PKI fits well the security requirements of the opportunistic mobile applications that rely on content forwarding over multiple hops. Hence, it can complement the existing security mechanisms, used to secure traditional always-connected mobile clients and provide a comparable level of security to opportunistic applications with the intermittent connectivity. However, complex cryptographic operations that are a part of the PKI based security, such as signature verifications, introduce non-negligible processing delays. We

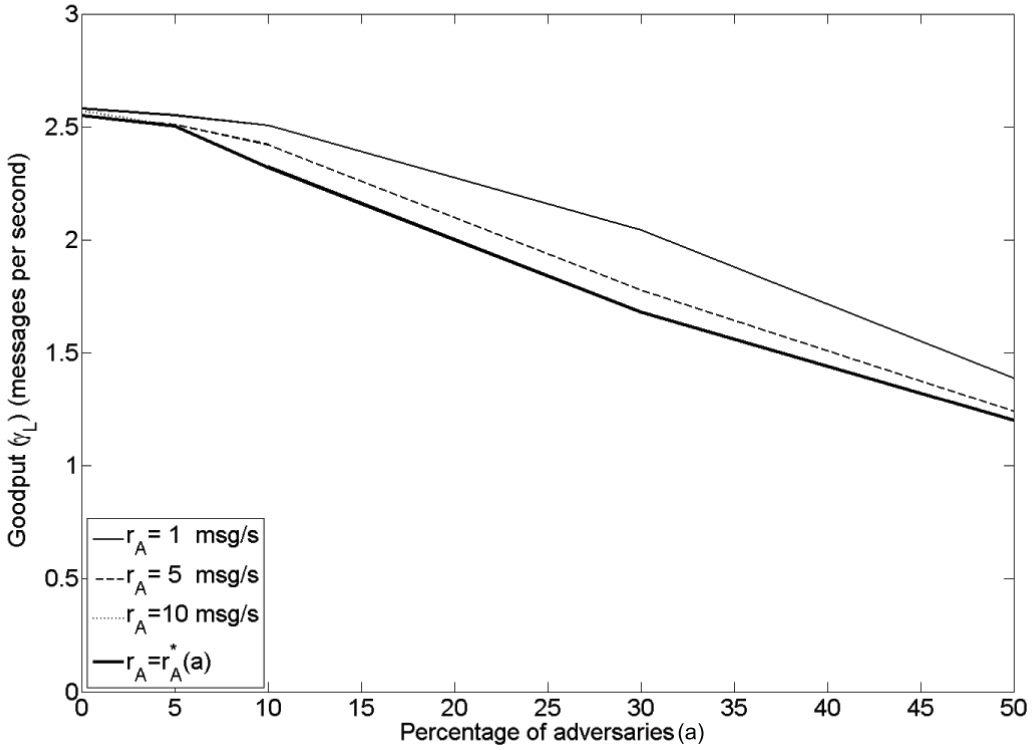


Figure 7.7: Goodput γ_L obtained for various adversarial sending rates r_A under pessimistic AMA for CGGC routing algorithm. The thick line corresponds to the optimal sending rate r_A^* selected by the adversaries.

show that the processing delays measured on the state-of-the-art platforms lead to performance degradation of the considered vehicular and opportunistic applications. Nevertheless, we also show that completely ignoring security verifications in favor of performance can lead to even a more severe performance decrease, due to DoS attacks.

We demonstrate that a simple, yet adaptive, filtering scheme that allows nodes to judiciously decide when to check the received messages that require further relaying, and when to simply forward them without any delay, significantly improves the performance of the applications that rely on multi-hop forwarding. The scheme, which we dub AMA, treats multi-hop messages in a reactive rather than a proactive way and requires checking of the relayed messages only in the presence of a threat. Our simulations with the state-of-the-art geocast routing algorithms demonstrate that, as a result of this security footprint reduction, the goodput of legitimate users increases up to 33%.

Closing Remarks and Complementary Material

Chapter 8

Conclusions

In this thesis we study the added value the opportunistic networks can bring to legacy networks in terms of content dissemination, but also with respect to energy saving and traffic offloading during the peak usage hours. Additionally, we propose a security framework and a security print reduction scheme for the class of opportunistic application clients that sporadically synchronize with existing web services. We base our findings on modeling, simulation with large data sets and experiments with live users and real applications.

We show that performance in different parts of an opportunistic network can be captured through a drift and jump model that takes into account a coarse-grained user mobility, contacts with infrastructure and contacts among mobile users. We demonstrate how the approximation of the model for large N can be used for optimal placement of infrastructure with respect to different utility measures.

Further, we show how opportunistic bandwidth can be exploited in combination with cellular bandwidth to offload a part of delay-tolerant 3G traffic during the peak usage hours. Using a large data set, we explore the roles of mobility prediction, opportunistic transfers and limited infrastructure placement. We reveal the relationship between delay and the amount of infrastructure needed to offload a certain volume of traffic, and we quantify the energy savings coming from the use of opportunistic bandwidth.

In addition to this, we demystify discrepancies in the conclusions of the previous studies about the utility of infrastructure in an opportunistic network, *i.e.,* we show that they are mostly due to the unrealistic assumptions about the cache sizes in mobile nodes. For this purpose, we implement a testbed with a real opportunistic application and live users. The experiment we perform also allows us to explain the significant differences between network performance

measured through simulation and the performance obtained experimentally. We show that a range of typically ignored factors in the simulation studies of opportunistic networks (limited contact duration, finite transmission bandwidth, finite cache sizes, interference and technology limitations) can completely alter the conclusions about the performance of a network.

The experiment further allows us to demonstrate how a statistical treatment of contact data sets using the weighted contact graph, offers a good prediction of certain performance measures (namely delivery ratios). We expose a strong dependency between a user centrality measure in this graph and the perceived delivery ratios and we fit a simple curve to this dependency, which persists with and without infrastructure. This means that it can be used to estimate the effects of adding infrastructure to an opportunistic network.

Finally, we design a security framework for the hybrid opportunistic applications that we target and a complementary adaptive scheme that protects the optimal relaying capacity in the opportunistic nodes. Using extensive simulations, we show that the scheme resists DoS attacks and yields significant performance increase.

Publications

Published

- N. Ristanovic, G. Theodorakopoulos, and J.-Y. Le Boudec, Traps and Pitfalls of Using Contact Traces in Performance Studies of Opportunistic Networks, *The 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*, Orlando, FL, USA, March 2012.
- N. Ristanovic, J.-Y. Le Boudec, A. Chaintreau, and V. Erramilli, Energy Efficient Offloading of 3G Networks, *The 8th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2011)*, Valencia, Spain, October 2011. (Best Student Paper Award)
- N. Ristanovic, P. Papadimitratos, G. Theodorakopoulos, J.-P. Hubaux, and J.-Y. Le Boudec, Adaptive Message Authentication for Multi-Hop Networks, *The 8th International Conference on Wireless On-demand Network Systems and Services (WONS 2011)*, Bardonecchia, Italy, January 2011.
- N. Ristanovic, D. K. Tran, and J.-Y. Le Boudec, Tracking of Mobile Devices Through Bluetooth Contacts, *ACM CoNEXT 2010 Student Workshop*, Philadelphia, PA, USA, November 30 - December 3 2010
- A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic, The Age of Gossip: Spatial Mean Field Regime, *The 11th International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance 2009)*, Seattle, WA, USA, June 2009. (Best Paper Award)
- N. Ristanovic, P. Papadimitratos, G. Theodorakopoulos, J.-P. Hubaux, and J.-Y. Le Boudec, Adaptive Message Authentication for Vehicular Networks, *The 6th ACM International Workshop on Vehicular Inter-Networking (VANET 2009)*, Beijing, China, September 2009.

In Preparation

- N. Ristanovic, and J.-Y. Le Boudec, Crowdsourcing Localization: Collaborative Tracking of Mobile Users Through Detectable Wireless Interfaces, *to be submitted to IEEE INFOCOM 2013*.

Curriculum Vitæ

Nikodin Ristanović
Laboratory for Computer Communications and Applications (LCA)
École Polytechnique Fédérale de Lausanne (EPFL)
1015 Lausanne, Switzerland

Education

2007–2012	PhD candidate in the School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland
2001–2006	Dipl. Ing. in Electrical Engineering and Computer Science, School of Electrical Engineering, University of Belgrade, GPA 9.71 (max 10)

Professional experience

2007–present	**Research and teaching assistant**, LCA laboratory, EPFL, Switzerland, TA for courses: TCP/IP Networking, Performance evaluation.
Summer 2010	**Research intern**, Thomson, Paris, France and Telefonica, Barcelona, Spain, Cloud assisted, energy efficient offloading of 3G networks.
2006–2007	**Core Network and Services Engineer**, Telenor, Serbia, Engineer in the Core Network and Services Planning Team of the mobile operator Telenor.
Summer 2006	**Intern**, Ericsson Services Ltd., DBA Team, London, UK, Maintenance and production of databases for the UK based 3G mobile operator Three.
Summer 2005	**Research intern**, Politecnica University, Madrid, Spain, Design and implementation of algorithms for segmentation of flying aircraft from images taken by thermovision camera.

Other skills and activities

TPC member	ACM S3 Workshop session chair, MobiHoc 2009, New Orleans, ACM S3 Workshop chair, MobiCom/MobiHoc 2010, Chicago.
Computer skills	Programming, databases, networking (Java, J2EE, C/C++, PHP, MySQL, JavaScript, Matlab, Linux, Android, Apache TomCat, ...)
Certificates	CCNA (Cisco Certified Network Associate), SND (Securing Cisco Network Devices).
Languages	English, French, Serbian, Spanish (basics).

Awards and honors

2011	*Best Student Paper Award*, MASS 2011, Valencia, Spain
2009	*Best Paper Award*, SIGMETRICS/Performance 2009, Seattle, WA, USA
2006	*Sreten Nedeljković Award for the top graduate*, School of Electrical Engineering, University of Belgrade, Serbia
2007	*The Scholarship of Excellence*, EPFL, Switzerland
2002 & 2005	*The Republic of Serbia Scholarship*, Serbia
2004 & 2005	*The City of Belgrade Scholarship*, Serbia
2003 & 2004	*The Scholarship of the National Foundation for Scientific and Artistic Youth Development*, Serbia

Bibliography

[1] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on the design of opportunistic forwarding algorithms," in *Proceedings of the 25th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2006)*, 2006.

[2] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic content distribution in an urban setting," in *Proceedings of the 2006 SIGCOMM Workshop on Challenged Networks (CHANTS '06.)*, 2006.

[3] P. Hui and A. Lindgren, "Phase transitions of opportunistic communication," in *Proceedings of the 3rd ACM Workshop on Challenged Networks (CHANTS '08.)*, 2008.

[4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary internet," *Communications Magazine, IEEE*, vol. 41, no. 6, pp. 128 – 136, June 2003.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*. ACM, 2003, pp. 27–34.

[6] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proceedings of the 25th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2006)*, 2006.

[7] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78 – 83, jan. 2004.

[8] 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 (MobiCom '06)*. ACM, 2006, pp. 334–345.

[9] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, ser. WSNA '02. ACM, 2002, pp. 88–97.

[10] 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," in *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*. ACM, 2002, pp. 96–107.

[11] A. Balasubramanian, Y. Zhou, W. B. Croft, B. N. Levine, and A. Venkataramani, "Web search from a bus," in *Proceedings of the second ACM workshop on Challenged networks*, ser. CHANTS '07. ACM, 2007.

[12] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. ACM, 2005, pp. 244–251.

[13] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, "Diversity of forwarding paths in pocket switched networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. ACM, 2007, pp. 161–174.

[14] N. Ristanovic, J.-Y. Le Boudec, A. Chaintreau, and V. Erramilli, "Energy efficient offloading of 3G networks," in *Proceeding of the 8th International IEEE Conference on Mobile Adhoc and Sensor Systems (MASS 2011)*, oct. 2011, pp. 202 –211.

[15] N. Ristanovic, G. Theodorakopoulos, and J.-Y. Le Boudec, "Traps and Pitfalls of Using Contact Traces in Performance Studies of Opportunistic Networks," in *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*, 2012.

[16] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic, "The age of gossip: Spatial mean field regime," in *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '09. ACM, 2009, pp. 109–120.

[17] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," *SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 145–158, Aug. 2004.

[18] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. ACM, 2005, pp. 237–243.

[19] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," in *SIGMOBILE Mobile Computing and Communication Review*, vol. 7, no. 3. ACM, July 2003, pp. 19–20.

[20] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *The First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004)*, October 2004, pp. 235 – 244.

[21] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," UCSD, Tech. Rep. CS-2000-06, 2000.

[22] A. Lindgren, C. Diot, and J. Scott, "Impact of communication infrastructure on forwarding in pocket switched networks," in *Proceedings of the 2006 SIGCOMM Workshop on Challenged Networks (CHANTS '06.)*, 2006.

[23] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: enhancing mobile networks with infrastructure," in *Proceedings of the 14th ACM international conference on Mobile computing and networking (MobiCom '08.)*, 2008.

[24] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnović, "Power law and exponential decay of inter contact times between mobile devices," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking (MobiCom '07.)*, 2007.

[25] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Enhancing interactive web applications in hybrid networks," in *Proceedings of the 14th ACM international conference on Mobile computing and networking (MobiCom '08.)*, 2008.

[26] N. Banerjee, M. Corner, and B. Levine, "An energy-efficient architecture for DTN throwboxes," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, May 2007, pp. 776 –784.

[27] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '08)*. ACM, 2008, pp. 241–250.

[28] S. Ioannidis and A. Chaintreau, "On the strength of weak ties in mobile social networks," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems (SNS '09.)*, 2009.

[29] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM 2005)*, 2005.

[30] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, 2006.

[31] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS '00)*, 2000.

[32] E. Altman, P. Nain, and J. Bermond, "Distributed storage management of evolving files in delay tolerant ad hoc networks," INRIA, Tech. Rep. 6645, 2008.

[33] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks*, vol. 51, no. 10, pp. 2867–2891, 2007.

[34] N. Eagle and A. Pentland, "Reality mining: Sensing complex social systems," in *Personal and Ubiquitous Computing*, 2005.

[35] E. Nordström, C. Diot, R. Gass, and P. Gunningberg, "Experiences from measuring human mobility using bluetooth inquiring devices," in *Proceedings of the 1st international workshop on System evaluation for mobile platforms*, ser. MobiEval '07. ACM, 2007, pp. 15–20.

[36] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[37] S. N. Diggavi, M. Grossglauser, and D. N. C. Tse, "Even one-dimensional mobility increases ad hoc wireless capacity," in *In Proceedings of the IEEE International Symposium on Information Theory (ISIT '03)*, 2003.

[38] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, p. 462, 2006.

[39] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the levy-walk nature of human mobility," in *The 27th IEEE Conference on Computer Communications (INFOCOM 2008)*, april 2008, pp. 924 –932.

[40] M. Gonzalez, C. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, 2008.

[41] H. Zang and J. C. Bolot, "Mining call and mobility data to improve paging efficiency in cellular networks," in *The 13th annual ACM international conference on Mobile computing and networking (MobiCom'07)*, 2007.

[42] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[43] J. Louhi, "Energy efficiency of modern cellular base stations," in *The 29th International Telecommunications Energy Conference (INTELEC 2007)*, Sept. 30 - Oct. 4 2007, pp. 475 –476.

[44] M. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Optimal energy savings in cellular access networks," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, june 2009, pp. 1 –5.

[45] K. Dufková and, M. Bjelica, B. Moon, L. Kencl, and J.-Y. Le Boudec, "Energy savings for cellular network with evaluation of impact on data traffic performance," in *2010 European Wireless Conference (EW '10)*, april 2010, pp. 916 –923.

[46] A. Rahmati and L. Zhong, "Context-for-wireless: context-sensitive energy-efficient wireless data transfer," in *Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys '07)*, 2007.

[47] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Trafficsense: Rich monitoring of road and traffic conditions using mobile smartphones," in *Microsoft Technical Report*, 2008.

[48] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Internet Measurement Conference (IMC '09)*, 2009.

[49] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath, "Combine: leveraging the power of wireless peers through collaborative downloading," in *Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys '07)*, 2007.

[50] T. Pering, Y. Agarwal, R. Gupta, and R. Want, "Coolspots: reducing the power consumption of wireless mobile devices with multiple radio interfaces," in *Proceedings of the 4th international conference on Mobile systems, applications and services (MobiSys '06)*, 2006.

[51] G. Ananthanarayanan and I. Stoica, "Blue-fi: enhancing wi-fi performance using bluetooth signals," in *Proceedings of the 7th international conference on Mobile systems, applications and services (MobiSys '09)*, 2009.

[52] H. Wu, K. Tan, J. Liu, and Y. Zhang, "Footprint: cellular assisted wi-fi ap discovery on mobile phones for energy saving," in *Proceedings of the 4th ACM international workshop on Experimental evaluation and characterization*, ser. WINTECH '09. ACM, 2009, pp. 67–76.

[53] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta, "Wireless wakeups revisited: energy management for voip over wi-fi smartphones," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, ser. MobiSys '07. ACM, 2007, pp. 179–191.

[54] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless: an event driven energy saving strategy for battery operated devices," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, ser. MobiCom '02. ACM, 2002, pp. 160–171.

[55] "Haggle - trust and security," http://www.haggleproject.org/deliverables/D4.1_final.pdf.

[56] "Haggle - trust and security," http://www.haggleproject.org/deliverables/D4.2_final.pdf.

[57] "Haggle - trust and security," http://www.haggleproject.org/deliverables/D4.3_final.pdf.

[58] "Security in Haggle Arhitecture," http://code.google.com/p/haggle/wiki/HaggleSecurity.

[59] J. Solis, "Securing network resources in opportunistic and delay-tolerant networks," PhD Dissertation, University of California, Irvine.

[60] K. E. Defrawy, J. Solis, and G. Tsudik, "Leveraging social contacts for message confidentiality in delay tolerant networks," in *Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference - Volume 01*, ser. COMPSAC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 271–279.

[61] J. Solis, P. Ginzboorg, N. Asokan, and J. Ott, "Best-effort authentication for opportunistic networks," *IEEE International Performance Computing and Communications Conference*, vol. 0, pp. 1–6, 2011.

[62] L. Lilien, Z. H. Kamal, V. Bhuse, and A. Gupta, "Opportunistic networks : The concept and research challenges in privacy and security," *Proceedings of the International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN '06)*, March 2006.

[63] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52 – 64, jan.-march 2003.

[64] M. Cagalj, S. Capkun, and J.-P. Hubaux, "Key agreement in peer-to-peer wireless networks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 467 –478, feb. 2006.

[65] N. Asokan, K. Kostiainen, P. Ginzboorg, J. Ott, and C. Luo, "Towards securing disruption-tolerant networking," Nokia Research, Tech. Rep. NRC-TR-2007-007, 2007.

[66] M. Gerlach, A. Festag, T. Leinmüller, G. Goldacker, and C. Harsch, "Security architecture for vehicular communication," in *WIT 2005*, Hamburg, Germany, 2005.

[67] IEEE1609.2, "IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages," July 2006.

[68] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure Vehicular Communication Systems: Design and Architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[69] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proceedings of the Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.

[70] P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "Securing Vehicular Communications - Assumptions, Requirements and Principles," in *Embedded Security in Cars (ESCAR '06)*, 2006.

[71] "Car2Car Communication Consortium Manifesto," http://www.car-2-car.org/.

[72] K. Laberteaux and Y.-C.Hu, "Strong vanet security on a budget," in *Workshop on Embedded Security in Cars (ESCAR '06)*, 2006.

[73] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in vanet," in *ACM VANET*, Montreal, Quebec, Canada, 2007.

[74] F. Kargl, E. Schoch, B. Wiedersheim, and T. Leinmüller, "Secure and Efficient Beaconing for Vehicular Networks," in *ACM VANET, short paper*, Sept. 2008.

[75] E. Schoch and F. Kargl, "On the efficiency of secure beaconing in vanets," in *The 3rd ACM Conference on Wireless Network Security (WiSec '10)*, 2010, pp. 111–116.

[76] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "On the performance of secure vehicular communication systems," *IEEE Transactions on Dependable and Secure Computing*, 2010.

[77] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *Proc. of ACM PODC*, 1987.

[78] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal multicast," *ACM Trans. Computer Systems*, vol. 17, no. 2, pp. 41–88, 1999.

[79] T. Bonald, L. Massoulié, F. Mathieu, D. Perino, and A. Twigg, "Epidemic live streaming: optimal performance trade-offs," in *Proceedings of the 10th international conference on Measurement and modeling of computer systems (SIGMETRICS '08)*, 2008.

[80] T. G. Kurtz, *Approximation of Population Processes*. SIAM, 1981.

[81] http://cabspotting.org/.

[82] IEEE1609.1, "IEEE trial-use standard for wireless access in vehicular environments (WAVE) - resource manager," 2006.

[83] A. Federgruen and H. Groenevelt, "The greedy procedure for resource allocation problems: Necessary and sufficient conditions for optimality," *Oper. Res.*, vol. 34, no. 6, pp. 909–918, 1986.

[84] WSJ, "AT&T to Urge Customers to Take Data Traffic Off Wireless Network," http://online.wsj.com/article/SB10001424052748704541004574600381410694794.html.

[85] "New York Times. iPhones Overload AT&T Network, Angering Customers, September 2009."

[86] "Bloomberg: AT&T to Pay $1.93 Billion for Qualcomm Mobile Spectrum," http://www.bloomberg.com/news/2010-12-20/at-t-agrees-to-acquire-wireless-licenses-from-qualcomm-for-1-93-billion.html.

[87] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Internet Measurement Conference (IMC '07)*, 2007.

[88] T. Hammond, T. Hannay, B. Lund, and J. Scott, "Social bookmarking tools (i): A general review," *D-Lib Magazine*, vol. 11, no. 4, April 2005.

[89] B. Lund, T. Hammond, M. Flack, and T. Hannay, "Social bookmarking tools (ii): A case study-connotea," *D-Lib Magazine*, vol. 11, Apr. 2005.

[90] "CNNMoney.com: Billions For Wireless Networks, December 28, 2010," http://money.cnn.com/2010/12/28/technology/billions_for_wireless_networks/index.htm.

[91] "AT&T Expands Wi-Fi Pilot Project," http://www.att.com/gen/press-room?pid=18162 &cdvn=news&newsarticleid=30982&mapcode=consumer.

[92] E. Halepovic and C. Williamson, "Characterizing and modeling user mobility in a cellular data network," in *The 2nd ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, 2005.

[93] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, "Measuring serendipity: connecting people, locations and interests in a mobile 3G network," in *Internet Measurement Workshop (IMC '09)*, 2009.

[94] K. Church, J. Neumann, M. Cherubini, and N. Oliver, "Socialsearchbrowser: a novel mobile search and information discovery tool," in *Intelligent User Interfaces 2010*, 2010.

[95] R. Angelova, M. Lipczak, E. Milios, and P. Pralat, "Investigating the properties of a social bookmarking and tagging network," *International Journal of Data Warehousing and Mining*, vol. 6, no. 1, pp. 1–19, 2010.

[96] "Belfast Telegraph. Apple's Ping Rival to Facebook and Twitter Launches with 160m Users, September 2010."

[97] D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo, and P. Rodriguez, "Fair wlan backhaul aggregation," in *Proceedings of the 16th ACM international conference on Mobile computing and networking (MobiCom '10)*, 2010.

[98] L. Hu, J.-Y. Le Boudec, and M. Vojnovic, "Optimal Channel Choice for Collaborative Ad-Hoc Dissemination," in *Proceedings of the 29th IEEE Conference on Computer Communications (INFOCOM '10)*, 2010.

[99] L. Song and D. F. Kotz, "Evaluating opportunistic routing protocols with large realistic contact traces," in *Proceedings of the 2nd ACM Workshop on Challenged Networks (CHANTS '07.)*, 2007.

[100] Guardian, "Voice-to-tweet," http://www.guardian.co.uk/technology/2011/feb/01/google-twitter-egypt.

[101] N. Y. Times, "Shadow Internet," http://www.nytimes.com/2011/06/12/world/12internet.html_r=3&hp.

[102] M. Weigle, "Improving confidence in network simulations," in *Proceedings of the Winter Simulation Conference (WSC '06)*, 2006.

[103] G. Lazarou, V. Frost, J. Evans, and D. Niehaus, "Using measurements to validate simulation models of tcp/ip over high speed atm wide area networks," in *Communications, IEEE*, 1996.

[104] J. Lawrence and T. Payne, "Exploiting familiar strangers: Creating a community content distribution network by co-located individuals," in *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web (FOAF '04)*, September 2004.

[105] Y. Richter, E. Yom-Tov, and N. Slonim, "Predicting customer churn in mobile networks through analysis of social groups," in *2010 SIAM International Conference on Data Mining (SDM 2010)*, 2010, pp. 732–741.

[106] E. Nordström, P. Gunningberg, and C. Rohner, "A search-based network architecture for mobile devices," January 2009, tR 2009-003, Uppsala University.

[107] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978-1979.

[108] "OAuth Authorization Protocol," http://oauth.net/.

[109] Aqualab, "Car-to-car cooperation for vehicular ad-hoc networks," http://www.aqualab.cs.northwestern.edu/projects/C3.html.

[110] T. Shinkawa, T. Terauchi, T. Kitani, N. Shibata, K. Yasumoto, M. Ito, and T. Higashino, "Technique for information sharing using inter-vehicle communication with message ferrying," in *7th International Conference on Mobile Data Management*, Nara, Japan, May 9-13 2006.

[111] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," in *Proceedings of the 7th International Conference on ITS Telecommunications*, Sophia Antipolis, France, June 2007.

[112] D. Page, D. J. Bernstein, and T. Lange, "Report on ebats performance benchmarks," European Network of Excellence in Cryptology, Tech. Rep. IST-2002-507932-D.VAM.9, March 2007.

[113] "ns-3 Wireless Network Simulator," http://www.nsnam.org/.

[114] "SWANS Wireless Network Simulator," http://jist.ece.cornell.edu/.

[115] "UULM extensions for SWANS simulator," http://vanet.info/node/12.

[116] C. Maihöfer, "A survey on geocast routing protocols," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 2, 2nd quarter issue 2004, IEEE.

[117] T. Leinmüller, C. Maihöfer, E. Schoch, and F. Kargl, "Improved Security in Geographic Ad Hoc Routing Through Autonomous Position Verification," in *ACM VANET*, 2006.

[118] A. Festag, P. Papadimitratos, and T. Tielert, "Design and performance of secure geocast for vehicular communication," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2456–2471, Jun 2010.

[119] F. Dötzer, T. Kosch, and M. Strassberger, "Classification for traffic related inter-vehicle messaging," in *5th IEEE International Conference on ITS Telecommunications (ITST)*, Brest, France, June 2005.

[120] N. Shibata, T. Terauchi, T. Kitani, K. Yasumoto, M. Ito, and T. Higashino, "A method for sharing traffic jam information using inter-vehicle communication," in *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*, San Jose, California, July 2006.

[121] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *Network, IEEE*, vol. 15, no. 6, pp. 30 –39, nov/dec 2001.

[122] C. Maihöfer, R. Eberhardt, and E. Schoch, "CGGC: Cached greedy geocast," in *WWIC*, Frankfurt/Oder, Germany, February 4-6 2004.

[123] H. Fuessler, J. Widmer, H. Kaesemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad-hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, 2003.

[124] http://www.denso-europe.com/.

[125] "Crypto++ Library," http://www.cryptopp.com/benchmarks.html.

[126] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in vanets," in *ACM VANET '04*. New York, NY, USA: ACM Press, 2004, pp. 29–37.

[127] "SUMO - simulation of urban mobility," http://sumo.sourceforge.net.

[128] M. Piórkowski, M. Raya, A. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "Trans: realistic joint traffic and network simulator for vanets," *SIGMOBILE MC2R*, vol. 12, pp. 31–33, Jan. 2008.

[129] "TraNS: open source tool for simulation of vanet applications," http://trans.epfl.ch/.