ABSTRACT

Title of dissertation: CONNECTIVITY
AND DATA TRANSMISSION
OVER WIRELESS MOBILE SYSTEMS

Nikolaos Frangiadakis,
Doctor of Philosophy, 2012

Dissertation directed by: Professor Nick Roussopoulos
Department of Computer Science

We live in a world where wireless connectivity is pervasive and becomes ubiquitous. Numerous devices with varying capabilities and multiple interfaces are surrounding us. Most home users use Wi-Fi routers, whereas a large portion of human inhabited land is covered by cellular networks. As the number of these devices, and the services they provide, increase, our needs in bandwidth and interoperability are also augmented. Although deploying additional infrastructure and future protocols may alleviate these problems, efficient use of the available resources is important.

We are interested in the problem of identifying the properties of a system able to operate using multiple interfaces, take advantage of user locations, identify the users that should be involved in the routing, and setup a mechanism for information dissemination. The challenges we need to overcome arise from network complexity

and heterogeneousness, as well as the fact that they have no single owner or manager.

In this thesis I focus on two cases, namely that of utilizing in-situ WiFi Access Points to enhance the connections of mobile users, and that of establishing Virtual Access Points in locations where there is no fixed roadside equipment available. Both environments have attracted interest for numerous related works. In the first case the main effort is to take advantage of the available bandwidth, while in the second to provide delay tolerant connectivity, possibly in the face of disasters. Our main contribution is to utilize a database to store user locations in the system, and to provide ways to use that information to improve system effectiveness. This feature allows our system to remain effective in specific scenarios and tests, where other approaches fail.

CONNECTIVITY AND DATA TRANSMISSION OVER WIRELESS MOVILE SYSTEMS

by

Nikolaos Frangiadakis

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2012

Advisory Committee:
Professor Nick Roussopoulos, Chair/Advisor
Professor Armand Makowski
Professor Ashok Agrawala
Professor Amol Despande
Professor Peter Keleher

Dedication

To my beloved parents Miranta and Manolis, and my equally beloved aunt, Elleni.

Acknowledgments

First and foremost, I would like to thank my advisor, Nick Roussopoulos, It is an honor to have been his Ph.D. student. I would like to thank him not only for the guidance he provided me, but also for his enormous patience. Thanks to his trust and support, I had the opportunity to implement the complex systems described in this thesis, and propose ideas some of which I hope will find their way into protocol implementations. He has had many students, and all I have spoken to are in agreement that it is a pleasure to work with and learn from such an extraordinary individual.

I would also like to thank my committee members, Dr. Armand Makowski, Dr. Peter Keleher, and especially Dr. Ashok Agrawala and Dr. Amol Deshpande that have been in my thesis proposal, and with whose advice this thesis was significantly improved.

It is with great pleasure that I thank here Dr. Daniel Camara. Our interest have been common through our scientific careers and without his contribution to our common work on Virtual Access Points while in EURECOM, this thesis would have been significantly lacking. It is impossible to count the nights we spent trying to improve our work.

Many thanks to Danila Kuklov. The initial version of PEGASUS system was part of our common project for Nicks class, and significantly contributed by both

keeping in mind the system potential, and implementing parts of it. He is also PEGASUS godfather, since he proposed the systems name when we were looking for one long past midnight hours.

The quality of this thesis and most related publications largely depends on the careful proofreading of Dr. Kostas Limniotis. Ever from undergrad years, he has proven a true friend, and a true scientist. His remarks and suggestions has helped me focus and improve my work.

I equally feel privileged to be able to thank Dr. Stergios Stergiou for being a true friend ever through undergrad. With both our discussions, and arguments, we have managed to grow as scientists. Stergio thank you, I wouldn't have it any way either!

I feel it is important to thank my colleagues at CNL in University of Athens, since our joint work there has helped me build a background in research. I would especially like to help Dr Stathes Hadjiefthymiades and Dr. Miltos Kyriakakos. I would like to thank Dr Panagiotis Stamatopoulos for his excellent work at teaching system programming.

Last, but not least, I would like to thank my dear friends I met in University of Maryland, Konstantinos Koutrolikos, Dr. Kostantinos Bitsakos, Dr. Antonis Deligiannakis, Dr. Dimitris Tsoumakos, Alexandros Tzannes, Kleoniki Vlachou, Thodoris Rekatsinas, Dr. Grecia Lapizco, Dr. Tsz-Chiu Au, Dr. Chris Papadopou-

los, Coddy Dunne, Dr. Dimitris Simopoulos, Dr. Elena Zheleva, Dr. George Theodorakopoulos, Dr. Iulian Neamtiu, Dr. Konstantinos Spiliopoulos, Anastasia Voulgaraki, Laura Kolias, Dr. Martin Paraskevov, Samuel Huang, to name a very small and very incomplete sample, for being there when I needed you, offering advice, spending your nights on common projects, and generally making my life in Maryland worth living. I would especially like to thank all the members of Digenis. It is impossible to remember all, and I apologize to those I've inadvertently left out.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

This chapter serves as an introduction to the thesis. As such it aims to present the problems examined and the common theme behind them. The importance of taking into account user mobility and its characteristics while designing wireless protocols is shortly explained. Since the existing protocols cannot efficiently support user mobility in their present form, we present in brief the ways we amend this problem which is the main subject of this thesis. Finally, the methodology adopted for our contributions and our scope is discussed.

Throughout this dissertation we focus on the effects of mobility to vehicular wireless networks. Our main goal is to show the need for a level in the system that tracks mobile users, and uses the information it gathers to improve network efficiency. Varying environments and networks are examined, and systems are proposed that include such a level resulting in contributions further explained in 1.5. In all developed systems this is one novelty that separates them from similar efforts. Furthermore PEGASUS is the first system able to operate over wide areas, transparently to users, over "in situ" networks. One of the novelties of the Virtual Access

Points (VAPs) system presented on Chapter 6 is that it leverages the user location information gathered to define areas and sets of users for VAPs.

1.1 Background

To better understand the problems presented, we next provide a brief description of the status of the protocols and systems later discussed. Note that is a sort introduction to the problems, and the related works are discussed later as part of Chapter 2. We see that, in today's world, mobile devices with multiple interfaces and varying capabilities are needed, and ubiquitous connectivity is expected by the users. However, the Internet is not designed with these standards in mind. Furthermore, even most of the wireless protocols used today are not designed for mobile devices with multiple interfaces and ubiquitous connectivity.

1.1.1 Interfaces

It should be stressed that Wi-Fi, Bluteooth, GSM, 3G, 4G, and possibly WiMaX are the most widely used protocols. Wi-Fi denotes the IEEE 802.11 family of standards and is currently mainly used for Personal Area Networks (PAN), and Local Area Networks (LAN) and very limited Wide Area Networks (WAN). Wi-Fi and Bluetooth operate on unlicensed spectrum, and so can be deployed by anyone.

All other protocols mentioned are typically operating in licensed spectrum, with Base Stations by a provider and have much longer range. WiMaX deployments operate on licensed or unlicensed spectrum.

The term xG, where x a number, refers to capabilities and specifications rather than a specific protocol. Thus GPRS end EDGE would be considered 2.5G, UMTS would is 3G, CDMA2000 and EVDO are very near 4G, but even the First release Long Term Evolution (LTE) and WiMaX do not completely satisfy the 4G requirements.

Bluetooth is today mainly used for lower short range communications from 5 to 10 meters. Typical 802.11g devices might have roughly a range of 35 m indoors and 140 m outdoors while 802.11n can be rated to exceed these ranges by more than two times. All other mentioned protocols range depends on the deployed Base Station's power and antenna, as well as the provisioning of the provider as mentioned later on 1.1.3. Note that we refer to ranges and capabilities of typical, "off the shelf" devices since for example even class 2 (10m) Bluetooth radios could be extended to 1.76 Km with directional antennas and signal amplifiers.

Less widespread interfaces include ZigBee (low power - mainly used in sensor networks) and Dedicated Short-Range Communications (used for Intelligent Transportation Systems - ITS).

An important point is that while GSM (considered as a second generation or

"2G") is a circuit switched network optimized for voice, all others are developed with data transmission in mind. In fact, modern and future standards are "all-IP", meaning that one network transports all information and services (voice, data, and all sorts of media such as video) by encapsulating these into IP packets. IP has become the pervasive, de facto standard. In today's word, users are almost always connected, and we are moving towards a future of ubiquitous connectivity where the wireless interface(s) used by the device will be transparent.

1.1.2 Mobile Devices and Applications

Wireless devices are pervasive in today's word. Most home users use Wi-Fi routers, and in fact global sales of Wi-Fi routers, access points and gateways exceeded 18 million units in the second quarter of 2010 [Res11]. Over 97% of notebooks today come with Wi-Fi as a standard feature, and an increasing number of handhelds and Consumer Electronics (CE) devices like games consoles and cameras are adding Wi-Fi capabilities [wif10]. Smartphone usage is rapidly increasing and should be expected to dominate the market. A new market for portable devices has appeared and some predict they might even replace Portable Computers as we knew them. Most of these smartphones and devices offer a combination of Wi-Fi, 4G, WiMaX, and Bluetooth. Even more, wireless devices are deployed in order to be used for specific applications like system monitoring, safety, and toll collection.

The existence of all these devices has resulted in a market for applications. User devices are often equipped with a vast number of sensors such as microphones, cameras, accelerometers, magnetometers, GPSs, and voltage meters. The enhanced sensing capabilities allow for the introduction of new services. Even further opportunities can result from the cooperation among large numbers of users or devices, possibly leveraging Social Networking or other Internet services. As we consider the Internet, Cloud Computing should also be mentioned.

The new services will increase data traffic. In fact, with the media capable devices that feature high quality and resolution screens it is reasonable to expect even higher bandwidth requirements. This is a trend that has already started, and has forced the network providers to revisit their offers for "unlimited data" plans.

This means that four of the factors that affect the amount of traffic of wireless data, namely the amount of users with smartphones, the frequency they use their devices, the number of services / applications they use, and the bandwidth requirements of these applications are all expected to grow, some of them exponentially. Of course, as we move to newer standards and protocols, the available bandwidth and capabilities of wireless networks will also increase, however, because of physical and power constrains, not necessarily in the same rate [Bro08].

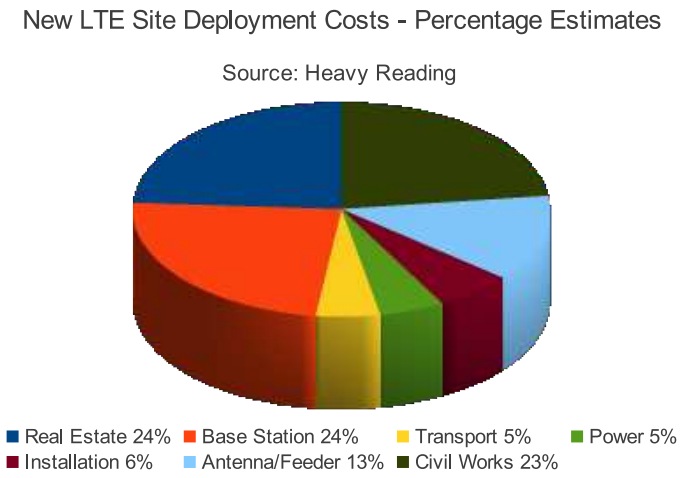

Figure 1.1: Deployment Cost

1.1.3 Cellular Infrastructure

Mobile Operators deploy Base Stations (BS) to allow their clients to connect to their network. BSs have a maximum range and the total available bandwidth that depends on the protocol and amount of spectrum used. Modern Base Stations use directional antennas to divide the area covered to sections and dynamically allocate bandwidth to active users. This means that at any given time, and depending on bandwidth requirements, only so many active users can be accommodated. In order to serve more users or increase the bandwidth available provided, the Providers decrease the range of the Base Station and reuse the available spectrum in a manner that avoids interference. Due to the complexity of the hardware, the cost of an LTE BS device is typically in the order of $100,000 to $200,000. However this is only a

fraction of the price since the Operator has to pay for the deployment of the BS and the Real Estate of the location. Figure 1.1 from [Bro08] shows an example estimated cost for deployment of an LTE BS. To cover really small areas, pico-cells and even femto-cells may be deployed, but the operator may still have to pay for deployment and maintenance. As bandwidth requirements and number of users increase, the operators will have to decrease the area of each Base Station, especially in densely populated areas, where the real estate is much higher. Also, the smaller the area covered by each BS, the less available options for placement are there, and the higher of the real estate cost. Therefore the total cost of the BS should be expected to further increase. Figure 1.2 from cellularmaps [cel10] shows the coverage of For the three of the largest US Operators in terms of area covered by 3G in 2011. Note that the speed on any covered area depends not only on the protocol used, but also on the number of other active users in the area and their traffic load.

1.1.4 Internet and deployed WAN Infrastructure

Most of the present and future applications make use of the Internet to store, retrieve information and contact other users. Mobile Operators traditionally connect to the Internet through Gateways. Wi-Fi Access Points (APs) can also allow Internet access to authorized users through the PANs, LANs, or even WANs they belong to. Wi-Fi APs are ubiquitous in urban environments as shown in Figure 1.3 which

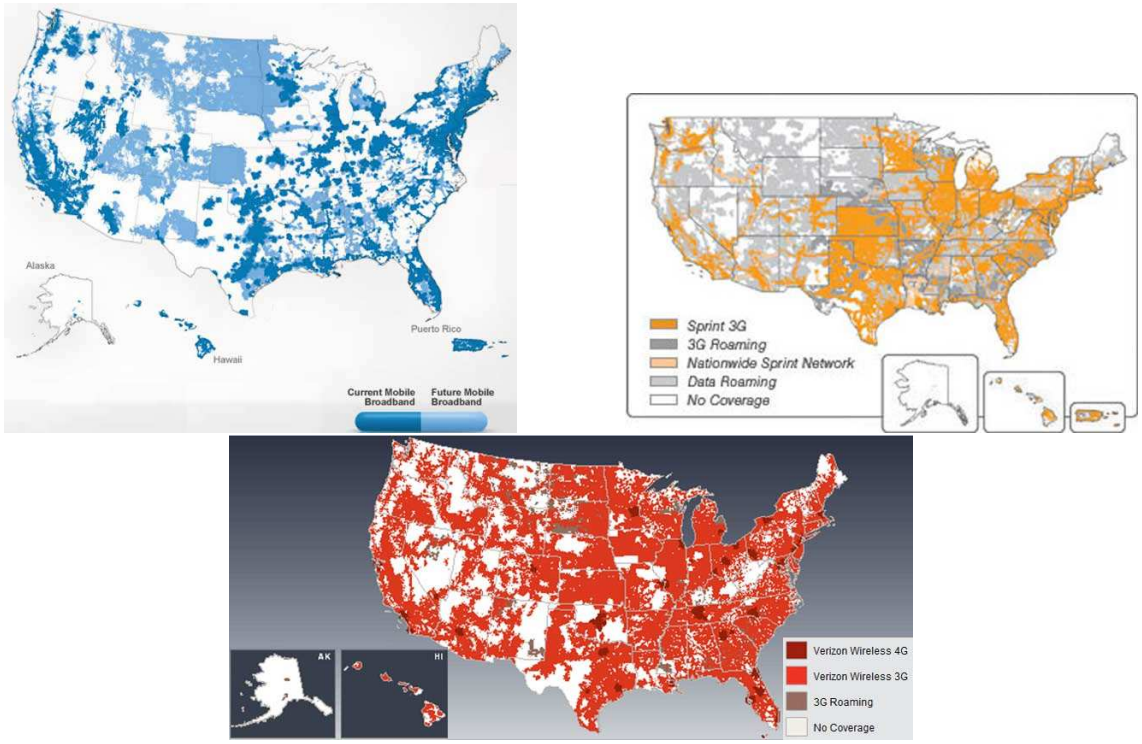

Figure 1.2: 3G and 3.9G Coverage

depicts the numbers of BSs and APs around the globe found while wardriving in 2010.

The police and some corporations have their own digital Professional Mobile Radio (PMR) systems of communications such as TETRAPOL and DECT. In many cases Intelligent Transportation Systems (ITS) could provide transport safety, productivity, travel reliability, informed travel choices, social equity, and road monitoring. It would be very beneficial to find how all this diverse infrastructure could be used to provide additional value services. Furthermore, in times of emergencies, when a disaster might affect a large part of the developed infrastructure, or even

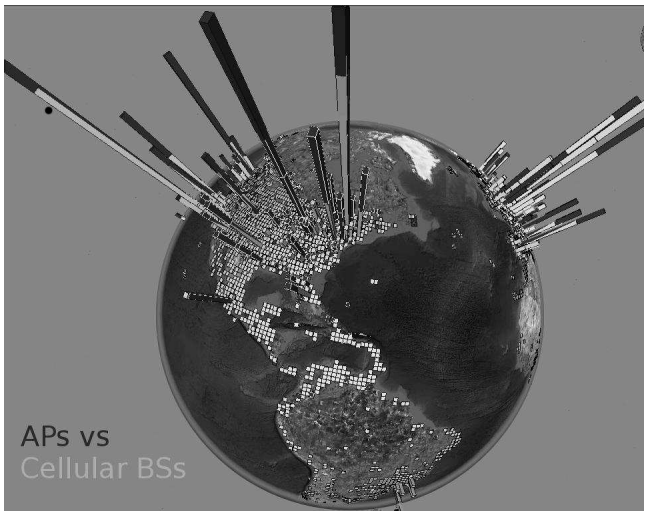

Figure 1.3: APs and Cellular BSs in 2010

some major event skyrockets traffic, it is crucial to be able to utilize this diversity to provide robust alternatives.

1.2 Problem Statement: Challenges

In this section we examine the problems that we encounter as a result of the existing infrastructure, devices and applications, and the future projections. We describe a number of problems and see that they have a common theme since the common reasons behind them are the effects of mobility and interaction of multiple systems. By examining these problems we can see how it is possible to contribute by composing more effective systems, and finding the principles that could be used allow us to design effective protocols for the future environments.

1.2.1 Devices with multiple interfaces and providers, on diverse networks

Future wireless systems will be pervasive and ubiquitous. They will have to operate over devices with multiple interfaces of varying characteristics. Any such system should allow the mobile user to connect with any and all available interfaces, decide how the connections should be made and how traffic should be routed. If we consider the current situation, we can see that there are huge margins of optimization in both the automation and the effectiveness of traffic routing. These challenges are so important that the inter-operability between Wi-Fi and WiMaX has been already taken into consideration. However there is no clear standard as to how a future wireless system will incorporate multiple interfaces belonging to different networks. In fact, in today's system each device has a different id for every network it connects. We should note that a large portion of the problem arises from the fact that the users are mobile, thus resulting in a dynamic environment. The cost and traffic characteristics of each available interface may vary depending on the location and the future path of the user.

As a response to this and other challenges, we propose PEGASUS, a system that extends the functionality of "in situ" Wi-Fi Access Points to enable wireless connectivity for fast moving vehicles. It can combine Wi-Fi to xG interfaces. We introduce proxies, track user location, and leverage global information of user move-

ment, as shown in Figure 1.6. As we discuss in Chapter 4, it provides clients with a constant IP address to preserve application sessions. Furthermore it is able to sustain connectivity in the absence of available APs by utilizing multiple physical interfaces. Efficient connection switching is achieved by storing a global DHCP connection to the PEGASUS server and predicting connection candidates on the client's path. A salient feature of the PEGASUS system is that it does not impose modifications to the infrastructure of deployed networks or protocols.

1.2.2 Varying requirements from applications

The purpose, scope and requirements of applications found in modern devices is very diverse. Some of these applications are interactive, necessitating low latency. Others produce bursty traffic, or support multimedia with large bandwidth requirements. A number of services can be critical, while others only demand best effort support. Some applications only require session connectivity by using short lived connections, or UDP. Others keep open TCP connections to the user device.

A system should allow for this diverse mix of applications. It should route traffic opting for the appropriate interfaces, depending on the location and mobility characteristics of the user. Moving one step further, a versatile system should allow us to combine networks used for different roles, with diverse specifications. As an example, consider allowing applications from PMRs, such as police radio, to run

over guest networks. In a disaster scenario, this might be the only available solution. Still, requirements of the applications, such as confidentiality and integrity, should be maintained. When the location of the user or the network conditions make it impossible to satisfy all application requirements, the system should degrade gracefully, offering at least best effort service, possibly keeping a channel for critical communications if it can do so, or resorting to delay tolerant delivery of important messages.

The range of possible requirements depends on the mix of applications in the system, hence, it is very difficult to design a single system that will effectively solve all these diverse challenges. However, throughout this thesis, we explore possible scenarios, and propose solutions. In many cases, we leave out of our scope addressing some security aspects that should be implemented in a real-world system implementation. We keep in mind how to implement security in the system, but leave it out of our prototype implementation.

Since cashing needs to be a part of any solution to these challenges, we examine how caching is affected by mobility with Mobile Cache in Chapter 3. We argue that existing analysis on caching becomes irrelevant for highly mobile environments. Mobile Cache dynamically adapts the bandwidth between push and pull according to both frequencies and the user mobility.

To address operating over guest networks, as well as best effort delay tolerant

delivery of important messages on uncovered areas, we introduce the Virtual Access Point (VAP) technique. As discussed in Chapter 6, VAPs transparently extend the reach of roadside access points to uncovered road areas. VAP nodes broadcast a series of objects, called program, based on their cache within the limits of a predefined region. This way, pushed information from VAPs does not interfere with AP traffic. In both PEGASUS and VAPs, we leverage global information of user movement. In this case we use the knowledge acquired in order to establish areas where the mobile users will act as virtual Access Points, as well as select the set of users that can act as VAPs.

1.2.3 Incorporating Infrastructure

One of the largest problems yet to be resolved is the ways that existing infrastructure will be coordinated to produce a powerful ubiquitous system. The capabilities of any wireless system depend on how effectively its mobile users can connect and take advantage of the vast Internet infrastructure, as well as other wireless networks. As wireless systems evolve, Internet connectivity becomes more effective. In parallel, the Internet infrastructure and services evolve to better accommodate mobile users. However, it is important to understand that the Internet Protocol was not originally designed with mobile users in mind. In fact, even the notion of what mobility means has evolved over the years. Once, a typical example

of a mobile user would be that of a user connected to his home, then disconnecting to go to work, or traveling to the airport with his laptop. Then upon arriving to a new location connecting again to the Internet. The effort was to allow this user to access most of his applications while offline. Today, a user should be expected to be almost always online, and the effort is to allow his applications unaffected by the bandwidth, latency and power constraints while he is on the move. This makes it even more important to focus on how the system can be efficient for mobile users.

It is said that "old protocols never die". In fact any future wireless system will probably evolve, and be able to take advantage of the current infrastructure. These protocols, as designed, cannot transparently and efficiently transfer data over multiple deployed mobile networks. In fact some were not originally designed for moving users, and others are not all-IP. However we should expect that some time will pass until all this infrastructure of significant cost is abandoned.

Both PEGASUS and VAPs are versatile solutions that allow us to incorporate existing infrastructure to extended mobile systems. PEGASUS takes advantage of "in situ" wireless networks to create an environment that will allow users to exploit their devices capabilities. It enables legacy as well as new applications and services, while being an efficient and scalable solution. VAPs extend connectivity to areas not covered by roadside APs, thus providing best-effort delivery of important messages to uncovered areas.

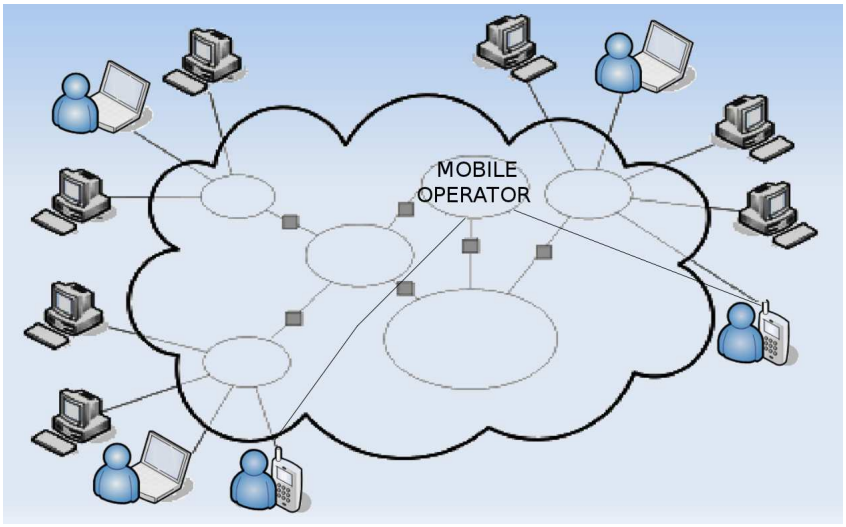

Figure 1.4: Mobile User connection examples

As a final note, Figure 1.4 shows an example of a typical situation of how a mobile user can connect to the Internet using his smartphone. We see that this particular user has two interfaces, one 4G, and one Wi-Fi. To use the 4G, he can subscribe with a Mobile Provider and use his network as a Gateway to the Internet. Alternatively, he can set his phone to detect and connect to a number of Wi-Fi APs, whenever his phone detects them in range. However, this situation is far from optimal and there is a lot of active research on how it can be improved. In section 1.3, we categorize these efforts and further explain the scope of our efforts to resolve this and similar problems.

1.3 Problem Statement: Classification

The main problem this thesis addresses is how location information can be used to efficiently organize large scale, heterogeneous vehicular networks of individual users, when these networks have no single provider.

For example, the devices presented in Figure 1.4, are organized to a system of networks. Currently, this system is not transparent. The user has to manually setup APs where the phone will connect. Furthermore, the two interfaces operate independently, meaning that there is no way defined to facilitate a cooperation between them. Note that Wi-Fi and 4G can complement each other in many ways as discussed in subsections 1.1.4 and 1.1.3. They have different power requirements while receiving, transmitting or being on stand-by mode.

In this section we examine the parameters of our problem so we can then present our solutions and the methodology behind them. The scale, structure, purpose and architecture of the network, are some of the major issues that will affect any design.

Figure 1.5 illustrates our focal area (in light gray), in relation to other approaches in the field (which are discussed in Chapter 2). Note that we condensed a lot of dimensions of the system to the "Environment Complexity" for illustration purposes. In fact, what we call here complexity, is a mix of factors, as discussed bellow.

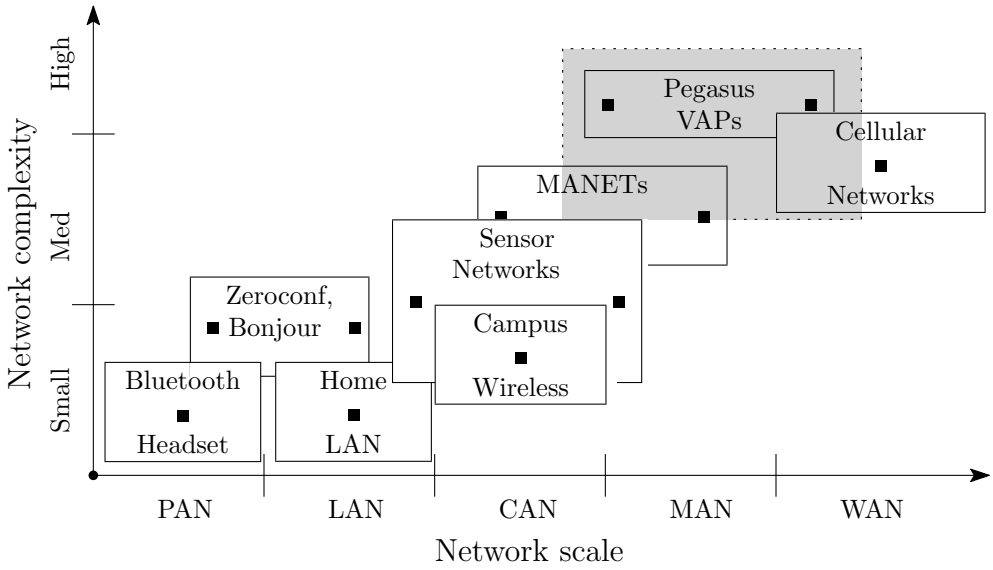

Figure 1.5: Wireless Network examples

1.3.1 Network Scale

Any network can be categorized depending on its scale. It can simply cover an area that extends just a few meters form a person and be used for communication among devices in proximity as is the case for personal area networks (PANs). It can cover a larger area, such as a house or lab, forming a local area network (LAN). There are also larger, campus area networks (CANs), and metropolitan area networks (MANs) which are usually limited to a campus or a specific metropolitan area respectively. Finally a wide area network (WAN) is a telecommunication network that covers a broad area and often links across metropolitan, regional, or national boundaries.

Depending on the scale of a network, the challenges we have to face to enable

efficient communication may vary up to a great extent. There is research going on, and protocols proposed all those types of networks. However, as the scale of the network increases, direct communication between all devices becomes impossible, and a set of interesting communication problems arise.

We are interested in the problems of how to efficiently identify the users, track their location, and setup a mechanism for information dissemination on MANs and WANs.

1.3.2 Network Complexity

The wireless network infrastructure may be owned by a single operator that can organize enforcing an efficient, often hierarchical structure, as is the case on 3G and 4G networks, or it can be comprised of a number of smaller parts with different owners.

It can be homogeneous, e.g., designed to work with devices that will all be equipped with UMTS, or allow for a variety of possible interfaces of the ones we show in section 1.1.1, each with its own characteristics. The network nodes might be all owned by a single entity and cooperate for tasks such as in the case of sensor networks. Alternatively, it may be owned by different users running a varying number of applications, each potentially acting as both a consumer and producer of data.

While some networks might provide a fixed infrastructure, there are cases where there is very limited or no infrastructure. In such cases, the nodes need to possess store and forward capabilities, and the network is delay tolerant.

We focus on heterogeneous networks of individual users, with no single owner. The requirements for efficient organization of such networks are being investigated. We address cases such as the one for PEGASUS, where ample fixed infrastructure is available, as well as others, where the present infrastructure is limited. In the later cases we introduce Virtual Access Points.

1.4 User mobility, tracking and optimization

In order to efficiently organize an architecture for a system of the complexity and scale of Figure 1.4, we will need to track the mobile users and devise a way to increase the system effectiveness based on our knowledge. Figure 1.6 illustrates one such architecture. In each case, we need to take into account the specifics of the environment we are studying.

We consider specific instances of vehicular networks. In the Chapters dedicated to each system, we demonstrate how the challenges mentioned in Section 1.2 apply to the specific environment of PEGASUS and VAPs. By leveraging global knowledge on user movement through a solution as the one of Figure1.6, we increase the effectiveness in mobile systems of both PEGASUS and VAPs.

For MobileCache and VAPs we use simulation because of the scale of the systems discussed. PEGASUS was implemented. The PEGASUS implementation allows us to demonstrate handoff times in practice. For scalability and measuring the effects of optimizing AP selection on PEGASUS we use simulation.

For the simulations, real traces were used whenever available. Real traces are often used in simulations in related work for measuring these systems efficiency and some of them are published. CRAWDAD[KHA], provides a large collection of user traces. However, the applicability of using a set of traces can depend on the trace characteristics and the characteristics of the system we are trying to analyze. To simulate the Mobile Cache system, we needed a large realistic mobility trace. We used the samples developed from LST [NBG06].

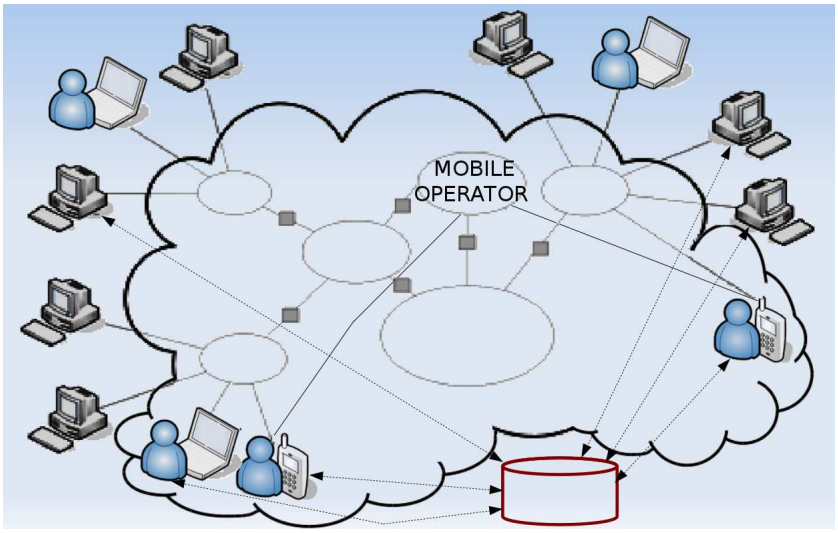

Figure 1.6: Tracking user mobility and utilizing proxies

1.5 Contributions

As we discussed in the previous section with Figure 1.6, one of our main contribution is to effectively utilize the system's knowledge of user movement. Moreover, with each system presented, we make multiple contributions.

MobileCache demonstrates one case where mobility significantly affects the efficiency of the wireless system and leads us to revisit our design principles. PEGASUS demonstrates an implementation where tracking mobile users and using them to optimize connectivity greatly improves the system's effectiveness. The simulations further show us the benefit of this approach. On VAPs we see how the same principle can be used to optimize a different, delay tolerant, ad-hoc network that can allow communications in case of disasters. Figure 1.6 depicts Figure 1.4 with the addition of a system to track part of the mobile users and utilizing fixed proxies to enhance connectivity.

1.5.1 MobileCache

In MobileCache, we consider information dissemination for mobile users and how caching should be implemented. We demonstrate that mobility significantly affects the efficiency of the wireless system and leads us to revisit our design principles. Therefore our contribution is twofold. First we provide a new analysis that takes into account the user mobility. We argue that in a highly mobile setting, anal-

ysis and optimization goals discussed in past papers become irrelevant, since the most important aspect of the system is not delay, but rather the ability to answer as many queries as possible. As we show, the optimal pull-push bandwidth ratio depends on the mobility patterns of the users. Second, we use our findings to build Mobile-Cache, a system that can efficiently answer multiple queries over a wireless environment. Our hybrid push-pull constitutes a very effective and scalable solution because it was designed taking into account user mobility.

1.5.2 PEGASUS

PEGASUS demonstrates how to optimize connectivity using user location information. Our system further shows how we can utilize an architecture similar to the principle presented in Figure 1.6 to address a complete array of the challenges in vehicular WLAN communications. PEGASUS provides wireless connection roaming at high velocities. To the best of our knowledge, it has been the first system to operate over "in situ" Wi-Fi networks, while at the same time offering transparency to user level applications by allowing a single IP address per user, and not impose additional requirements to existing infrastructures. PEGASUS offers simple deployment, improved scalability, and is the first able to operate over secure "in situ" networks. It remains efficient under intermittent connectivity conditions and supports heterogeneous network mediums for increased robustness.

With PEGASUS we see an implementation where the benefits of using data from tracking user mobility to improve efficiency become clear. Discussing the scalability, we see how further optimization is possible by utilizing predictions for user mobility to suggest connections for the mobile users. We see how suggesting the APs allows for longer average association times and relatively balanced load.

While implementing PEGASUS we published fixes to bugs in Click's DHCP module and ported Click to Android.

1.5.3 Virtual Access Points

On VAPs we see how the same principle can be used to optimize a different, delay tolerant, ad-hoc network that can allow communications in case of disasters.

With Virtual Access Points (VAPs), we introduce a new technique that allows data dissemination among vehicles to extend the reach of roadside access points to uncovered road areas. We use user mobility data to optimize the locations of VAPs and decide which set of vehicles have the potential to become effective VAPs based on their current direction of movement. Each vehicle that receives a message from an Access Point (AP) stores this message and rebroadcasts it into non covered areas. We show how this design can help us avoid interference since each operates on a bounded region outside any AP. We see how mobility data can allow us to improve the effectiveness of VAPs when used for divert applications such as stream based

traffic and broadcasting important traffic in disaster scenarios where VAPs allow the system to remain operational under conditions it would otherwise be inadequate.

1.5.4 Scope

This dissertation focuses on Vehicular users of Wireless Networks. We concentrate on how we can enhance the efficiency of the system by taking into account the mobility of the users.

The implementation of a system like the one described is beyond the scope of this dissertation. There are many and very complex problems that need to be addressed before the standards that would allow it can be drafted. Not all of them have to do with Computer Science.

Security is an important factor on any system that needs to be addressed from the first, designing phase. Ensuring safety, privacy, confidentiality, and accountability on any system is not trivial. However we will not address this problem.

Power consumption of the mobile devices is crucial on a number of wireless systems. We do not consider it, since users on the vehicles can tap into the vehicles power supply.

Decisions on pricing based on marketing, decisions on how to handle traffic between providers often resulting in legal constrains, social reasons for the success or failure of a system, support from corporations to different standard groups, po-

litical pressure, and incentives to join a system before it achieves critical mass are all important and valid problems that are not addressed here. Although a battle between standards like WiMaX and LTE might be decided on some of those, we will not address them.

Our problem is to consider the effects and importance of mobility on a variety of wireless vehicular networks with current interfaces, protocols and deployments, and try to decide on how it can be handled efficiently.

1.6 Roadmap

In Chapter 2 we examine related work for systems as well as work that produced results we used to develop our systems. Chapter 3 discusses the effects of mobility for caching and the MobileCache system, Chapter 4 the implementation and architecture of PEGASUS, Chapter 5 the scalability of PEGASUS and how mobility data can be used to optimize selection of future Access Points. On chapter 6 the VAP architecture is described, then the simulations performed under varying mobility and traffic are discussed. Then we conclude on Chapter 7

Chapter 2

Related Work

This chapter aims to explain the terms, technologies and methods used in later Chapters. Furthermore, it refers to recent developments related to this work. This thesis concentrates on Vehicular Networks, and so it is natural to focus a large portion of this work section in this direction. We examine traditional approaches, store and forward networks, as well as their hybrids. Some of the works examined refer to actual system implementations, some to simulations, and others to particular aspects of parts that are crucial the efficiency of either wireless technologies or the systems that equip the network and forward the data on top of them.

2.1 Link Layer

This work focuses higher than the Link Layer. However, the characteristics of the wireless technology that will be used will affect the design and effectiveness of the network. In this thesis we focus on WiFi, or WiFi combined with Cellular or WiMax. WiFi are often local area networks, APs can be deployed by anyone and penetration. Cellular and WiMax on the other hand, are deployed and maintained

by wireless providers and as such support user mobility. Below we provide tables with the range and bandwidth of different wireless technologies.

2.2 Transport Layer

2.2.1 Wireless Implementations of Networks for data dissemination providing connectivity

The performance of TCP and UDP in wireless network scenarios from immobile clients has been relatively well-studied [ABB+04]. However, not many research efforts attempted to characterize WLAN performance for moving vehicles. The Drive-thru Internet project by Ott and Kutscher [OK04b] studied the behavior of network connections over 802.11b and 802.11g from a moving car. The study involved a number of measurements over both UDP and TCP, and the goal was to understand the impact of the car's velocity, transmission rate, bit-rate, and packet size on throughput and delay. Ott and Kutscher classified WLAN connection period as three stages: the "entry" stage, "production" stage, and "exit" stage. During the entry and exit stages, the vehicle is far from the Access Point and throughput is low. However, when the distance is ≈200 meters from the Access Point, the connection is considered to be in the "production" stage. This is the stage where the significant volume of data can be transferred. Drive-thru project shares our

position to use intermediate proxies to further improve connection performance. In their more recent work [OK], they show that they can avoid TCP start up overheads by using proxies, and hiding short period of disconnection from the transport layer. In PEGASUS, instead of concentrating on modification of the usual TCP behavior, we concentrate on providing a constant connectivity appearance to the client, without the need to deal with re-initialization of the broken TCP connections. This is achieved by avoiding DHCP discovery costs during WLAN connection acquisition for fast and efficient connection transitions, and by providing a layer on top of the physical network cards to offer a persistent IP address to client applications.

Another study that demonstrated the feasibility of using off-the-shelf 802.11b wireless connectivity from a moving car was performed by Gass et al [GSD06]. The experiments where conduced in a controlled environment and they measured performance from a mobile client to a single access point in the California desert. The authors measured the connection quality between the client and the AP, and they concluded that packet losses are low within 150 meters of the access point for a wide speed range (5-75 mph).

While the two studies above demonstrate the possibility of using a wireless network from a moving car, more projects were carried out to study IP communications on the road. The FleetNet [URL] project investigates inter-vehicle communication in wireless add hoc network, for traffic-related control information using address-

ing geo-based routing. Similarly, the Hocman project [EJs02] also addresses data sharing across vehicles. An important method to upload data to the Internet via already deployed and open wireless 802.11b/g is proposed by MIT CarTel project [BHM+06]. The MIT group performed a study on the availability of open urban WiFi networks, and they attempted to estimate the performance of using "in situ" access points. The experiment involved several cars that were driven in the Boston and Seattle metropolitan areas. Their results exhibit an average connection time of 13 seconds to a single access point while driving, and their major challenge was to reduce connection setup times when the car exited one network and entered another one. In PEGASUS we concentrate not only on the case of the performance of a single client, but propose a complete system to support vehicular WiFi network connectivity. We look at all of the clients managed by a server as infrastructure with common resources and knowledge about access points. Furthermore, PEGASUS's global DCHP cache repository significantly improves connection switch efficiency and scalability.

Other numerous research activities worked on solutions to mitigate disruptive effects of handovers which cause intermittent connectivity in the mobile communication environment. Many of them suggest modifications in the transport protocol layer. I-TCP [BB95] is a split connection approach that introduces a transport layer intermediary for splitting a TCP connection between a fixed and a mobile host into

two connections. The idea is to isolate the fixed host from communication irregularities of the mobile host. I-TCP explicitly breaks the end-to-end semantics of TCP, i.e. TCP connections are terminated at the intermediary. In case of a handover, a state transfer from one I-TCP to another has to occur. The Snoop protocol [BSAK95] provides a more transparent support layer, and relies on a dedicated agent that "snoops" on the TCP communication on the path between the mobile and fixed station. It buffers TCP segments and offers retransmission services. In case of a handover a state transfer is not necessarily required. In our approach we choose not to modify the underlying TCP layer to enhance the TCP performance, but instead rely on the currently deployed infrastructures and protocols.

The projects described above are confined to using WiFi for all of network communication. Other systems like CAMA [BWLW04] and Mobile Router [RCC+04] explored using multiple wireless mediums. CAMA utilized cellular communications for control messaging purposes, while Mobile Router concentrated on allowing different client types (bluetooth, cellular, etc...) to connect to a common router on a commuter bus. The router would search for multiple available network types in the area for an outside connection as well. However they did not attempt to examine the mobility of the outbound connection issues, and concentrated on efficient ways to service the internal router network on the bus. In PEGASUS we allow every client to use multiple physical mediums whenever the client is capable of doing so,

and clients can transparently switch between mediums without changing IP.

2.2.2 Delay Tolerant Systems for Vehicular Networks

Delay Tolerant Networking, sometime referred to as Disruption Tolerant (DTN), has been developed as an approach to building architecture models tolerant to long delays and/or disconnected network partitions in the delivery of data to destinations. In this section, we will study the characteristics of these architectures, and many of the protocols developed to ensure packet delivery in these networks. We henceforth use DTN to refer to both Delay Tolerant Networking and Disruption Tolerant Networks. For Vehicular DTN, the acronym VDTN is used.

The vehicular network research field, and in extent the VDTN research field, have attracted great attention in the last few years. Initiatives such as i2010 Intelligent Car Initiative Intelligent Car (2009) aim to decrease the accidents and CO2 emissions in Europe utilizing sensors and vehicle-to-vehicle (V2V) communication to increase road safety. According to these projects, cars equipped with wireless devices will exchange traffic and road safety information with nearby cars and/or roadside units.

According to the ETSI 102 638 technical report (ETSI TR102_638, 2009, June), the 20% of the running vehicles will have wireless communication capabilities by 2017. The same report estimates that by 2027 almost 100% of the vehicles will

be equipped with communication devices.

The design of the core Internet protocols is based on a number of assumptions, including the existence of some path between endpoints, small end to end round-trip delay time, and the perception of packet switching as the right abstraction for end-to-end communications. Furthermore, the efficiency of these protocols is based on assumptions about the resources available to the nodes and the properties of the links between them. Traditionally nodes are considered to be fixed, energy unconstrained, connected by low loss rate links, and communication occurs due to the exchange of data between two or more nodes.

Today, however, new applications, environments and types of devices are challenging these assumptions and call for new architectures and modes of node operation. Some of these challenges are intermittent and/or scheduled links, very large delays, high link error rates, diverse and/or energy constrained devices, with heterogeneous underlying network architectures and protocols in the protocol stack, and most importantly, the absence of an end-to-end path from a source to a destination. Applications that may pose such challenges include spacecrafts, planetary/interplanetary, military/tactical, disaster response, mobile sensors, vehicular environments, satellite and various forms of large scale ad hoc networks. The variety of these applications, the impossibility of having a fixed wired Internet infrastructure everywhere, and the inclusion of mobility in most of these applications, make

these challenges more difficult to surmount. This often leads us to a new approach of designing networks, taking into account several constraints and characteristics, using DTN.

We continue to examine what is DTNs, their main advantages and disadvantages as well as some of the main research subjects that involve DTNs.

VDTNs have evolved form DTNs and are formed by cars and any supporting fixed nodes. Fall (2003) is one of the first authors to define DTN and discuss its potential. According to his definition, a DTN consists of a sequence of time-dependent opportunistic contacts. During these contacts, messages are forwarded from their source towards their destination. This is illustrated in Figure 1, in the first contact the origin sends the message to A in time t1, then A holds the message until it delivers to the destination in the contact at time t2. Contacts are characterized by their start and end times, capacity, latency, end points and direction. The routing algorithm can use these pieces of information to decide the most appropriate route(s) to deliver a message from its source to its destination. However, routing in a network where the edges among the mobile nodes depend on time signifies is not a straightforward task. One needs to find an effective route, both in time and space. All nodes along the path should consider the nodes movement pattern and the possible communication opportunities for message forwarding. Unfortunately, it is not always easy to determine future communication opportunities or even forecast

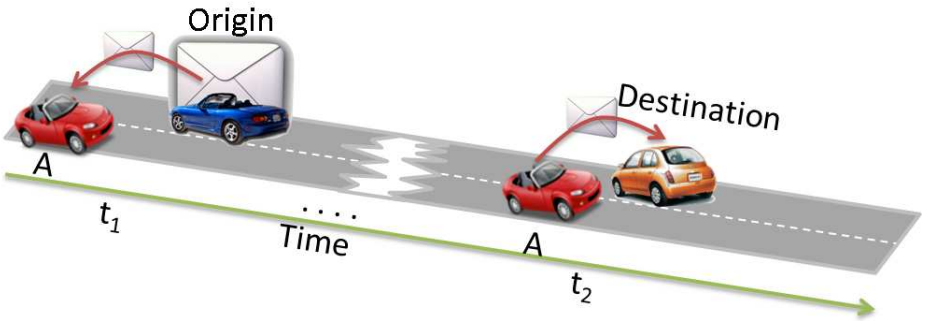

Figure 2.1: A representation of two communication opportunities in a VDTN

the mobility patterns of the nodes in the network.

Cerf et al. (2007) characterizes contacts as:

- Persistent: When they are always available, i.e. a Cable modem
- On-Demand: When they require an action to start, but after that they work as persistent contacts. For example, a dial-up connection.
- Intermittent scheduled: When the parts agree to meet at a specific location for a determined period of time, i.e., low earth satellite communication window.
- Intermittent opportunistic: Contacts that occur unexpectedly, for example, a car passing by in a non scheduled manner.
- Intermittent predicted: When the contacts are not based on a schedule, but on predictions. A prediction is a contact that is likely to happen, based on

the history or other kind of information. However, there are no guarantees predicted contacts will actually happen. For example, when people are commuting to work, it probable that at the same time the same contacts are available, because people normally go at the same time, take the same routes.

Forwarding and routing strategies may vary significantly according to the type of contacts a node, or a network, is expected to encounter. In the case of a DTN with intermittent contact opportunities, the main priority is to maximize the probability of message delivery, and to minimize the end-to-end delay. For networks with more stable and consistent contact opportunities, it is important to discover an efficient path while trying to save as much as possible of the network resources. On a scenario with deterministic message routing and persistent, on-demand or intermittent but scheduled contacts, we may have a chance to achieve optimal performance of the network and manage efficiently the available recourses, i.e. spectrum and node energy. However, under unpredictable intermittent network conditions, where the mobility obscures present and future topology, nodes can only forward packets randomly based on the likelihood they will eventually arrive to their destination. Then, the problem of delivering messages to their final destination is paramount and dominates that of resources utilization. In this case, flooding or epidemic message forwarding are popular approaches. Between the two extremes, deterministic contacts and fully opportunistic ones, a broad range of strategies may be used to

balance message delivery and resource optimization. Another issue to keep in mind is the duration and bandwidth of contacts. In a vehicular network, the number of contacts may be high, but the duration of each one should often be expected to last only seconds, especially between cars moving in opposite lanes. This significantly limits the amount of information exchanged among nodes.

In 2002 the Internet Research Task Force (IRTF), (IRTF, 2009), started a new group called Delay-Tolerant Networking Research Group (DTNRG), (DTNRG, 2009). The group was first linked to the Interplanetary Internet Research Group (IPNRG), (IPNRG, 2009), however, soon it became clear that the main characteristics of DTNs, i.e. non-interactive, asynchronous communication, would be useful in a broader range of situations. The main aim of IRTF is to provide architectural and protocol solutions to enable interoperation among nodes in extreme and performance challenged environments where the end-to-end connectivity may not exist. IRTF states DTNRG focuses to a wide range of application scenarios; spacecraft, military/tactical, public safety, underwater, sensor and ad-hoc networks and extremely degraded connectivity, such as country side networks, to name a few.

2.3 Challenges and Techniques

The conditions of DTN operation lead to an architecture that challenges the traditional conception of most of the network layers. In this section we present some

of the major challenges faced by DTN protocols at different network layers. Standard network modeling techniques are also challenged and new ways to model nodes and connections should be created to evaluate the considered protocols. Therefore this section also discusses the different network modeling, traffic modeling, transport layer issues, routing and data dissemination strategies.

2.3.1 Routing

The challenges that Delay Tolerant Networking (DTN) need to overcome have lead to significant research focused on routing. Routing is considered to be the problem of deciding forwarding strategies that enable messages to pass from the origin to the destination. The issues presented in this section pertain to most of the network layers and techniques developed for DTNs. For the case of VDTNs, store-and-forward, or store-carry-and-forward techniques are used (Small, Haas, 2005). This means that the nodes which receive a message, store it for some time, possibly carry it to another location, and afterwards forward it to other nodes. In the Internet, nodes often momentarily buffer packets as well. However, this spends as sort time intervals as possible, while in DTN it is used to overcome absence of end-to-end connectivity, as well as a mechanism to wait until efficient connections are present. Each intermediate node verifies the integrity of the message before forwarding it. In general, this technique helps us cope with intermittent connectivity, especially in

the wilderness or environments requiring high mobility, and may be preferable in situations of long delays in transmission and variable or high error rates.

Mundur, & Seligman (2008), identify mainly two classes of routing algorithms for DTNs. The first class is based on epidemic routing, in which nodes use opportunistic contact to infect other nodes with the message to be delivered. For this group, the need of network knowledge is minimal. The routing algorithms have no control of node mobility and the forwarding process occurs in a fortuitous way. The second class of algorithms utilizes topology information and the algorithms may control nodes mobility. For Mundur, & Seligman (2008), this case is characterized by "islands" of well connected nodes with intermittent connectivity with other nodes.

Fall (2003) and Jain, Fall & Patra, (2004) present an interesting list of routing issues for DTNs.

- Routing objective: Although the main objective of a routing algorithm is message delivery and DTNs are, by definition, tolerant to delay, that does not mean we should not try to decrease the delay as much as possible. Algorithms should attempt to find a good tradeoff between decreasing the end-to-end delay and saving network resources.

- Reliability: The protocols should be reliable and provide some form of mechanism to inform the nodes that their messages reached the destination. Acknowledged message delivery is an important enhancement of the offered ser-

vices.

- Security: In all types of networks, security is an important factor. However, in DTNs the packets may cross a diverse path to reach the destination. The reliability and intentions of often numerous intermediate nodes may not be possible to guarantee. Mechanisms to provide message authentication and privacy of the messages content are of supreme importance.

- Resource allocation: Normally the main routing objectives of maximizing the message delivery ratio and minimize resource allocation are conflicting. The easiest way to guarantee the message delivery in the smallest amount of time is flooding the network with the message. However, this means a high use of network bandwidth, nodes memory and processing power. These may lead to other problems such as packet collisions, packet drops because full message queues and surely the waist of the limited amount of energy of the nodes.

- Buffer space: Considering the disconnection problem, messages may be stored for a long period of time before they can be forwarded. The buffer space must to be enough to maintain all the pending messages, i.e. messages that did not reached their final destination yet.

- Contact scheduling: The forwarding waiting time is one of the principal elements on DTNs. It is not always clear how long a node will need to keep a

message to enable its forwarding. This period may vary from seconds to days.

- Contact capacity: Not only the contacts may not always be predicted, but when they occur, they may be brief. The protocols should take this into account and try to minimize, as much as possible, the use of the spectrum and time with control messages.

- Energy: Mobile nodes may have limited amount of energy and, possibly, hard access to power sources. Normally, for VDTNs the energy is a factor to be kept in mind but it is not one of the main factors since the vehicle can normally provide enough energy to maintain the communication system.

To evaluate the routing algorithms for DTNs Jones (2006), and Sanchez, Franck & Beylot (2007), propose the utilization of:

- Delivery ratio: Jones (2006) defines delivery ratio as "the fraction of generated messages that are correctly delivered to the final destination within a given time period"

- Latency: Even though the networks and applications are supposed to endure delays, many applications could take advantage of shorter delays. Even more, some application have time windows of delay resilience, i.e. messages are valid during a certain amount of time, after that the message looses its validity.

- Transmissions: The number of messages transmitted by the algorithms varies and some, that create multiple copies of the message, may send more messages than others.

- Lifetime: Route lifetime is the time a route can be used to forward packets without the need for re-computation.

- End-to-end delay: This evaluation criterion is the time it takes for one message to go from the origin to the destination.

- Capacity: Capacity is the amount of data that that may pass through one route during its lifetime

- Synchronicity: Even in a delay tolerant network, it is possible that, during some intervals, origin and destination are close and the communication may occur directly, or in the same way as it is in traditional wireless networks. Synchronicity, measures how long this situation where classical communication is possible.

- Simultaneousness: This criteria measures the contact durations. I.e. the time intermediate nodes are in the same area,.

- Higher order simultaneousness: Simultaneousness is the computed hop-by-hop. However, the same concept may be applied to a series of nodes, a k

subsequent number of nodes i.e. a segment of k consecutive nodes, that are part of the complete path.

- Discontinuity: is the normalized duration of packet storage thorough the path.

Recently, Shen, Moh & Chung (2008) presented a compact and interesting list of routing strategies for DTNs. Like Mundur, & Seligman (2008) Shen, Moh & Chung (2008) also divide the routing protocols in two families; flooding and forwarding. Flooding strategies are the ones where nodes create copies of the packet and forward to more than one node. Forwarding strategies use the knowledge of the network to select the best paths. A comparison between the generic behavior of flooding and forwarding strategies is depicted in Error: Reference source not found. Note that flooding strategies result in a significantly higher number of messages compared to forwarding.

Flooding based strategies

One of the simplest possible forwarding strategies is called Direct Contact. On this strategy the node waits until the source comes into contact with the destination before forwarding the data. Jones (2006) considers direct contact as a degenerate case of a forwarding strategy. Even though this strategy does not multiply messages, it is considered flooding. The reason is that it does not make use of any topology

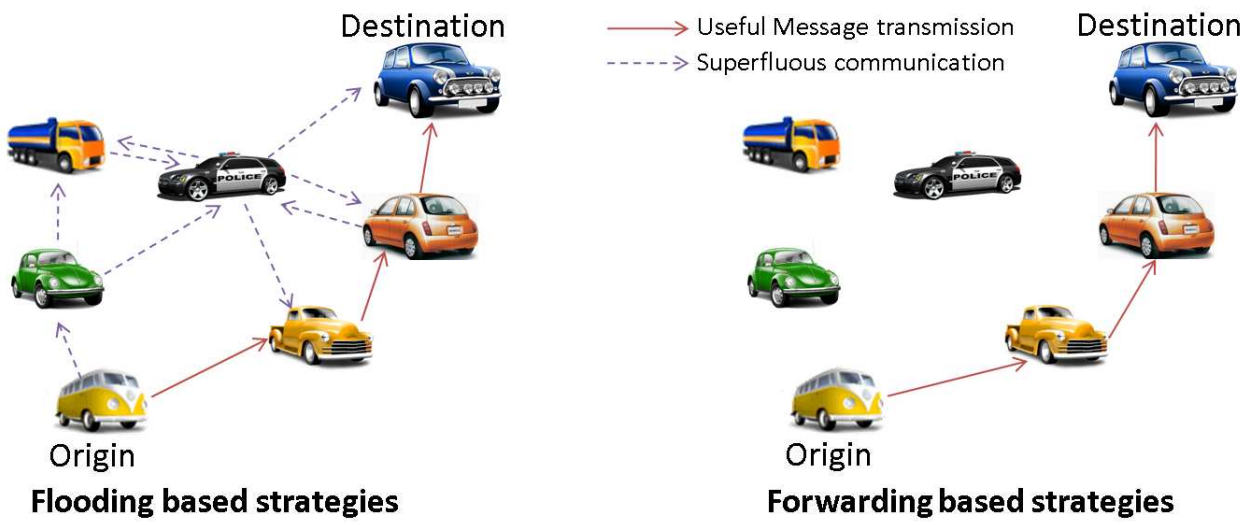

Figure 2.2: Message transmission example comparing Flooding based and forwarding based strategies.

information the nodes possess. The strategy is simple and presents low resource consumption; however, if the contact opportunities between source and destination are low, then the delivery rate can be low.

On the Two hop Relay strategy (Jones, 2006), the source copies the message to the first n nodes that it contacts. These nodes relay the message until they find the destination, it is similar to the direct contact, but now not only the source keeps the message, but also n copies of the message are spread among other nodes. With this we increase the required resources, but also the expected delivery ratio.

Three based flooding strategy (Jones, 2006), extends even more the idea of direct contact in the sense that now all nodes that receive the message may create

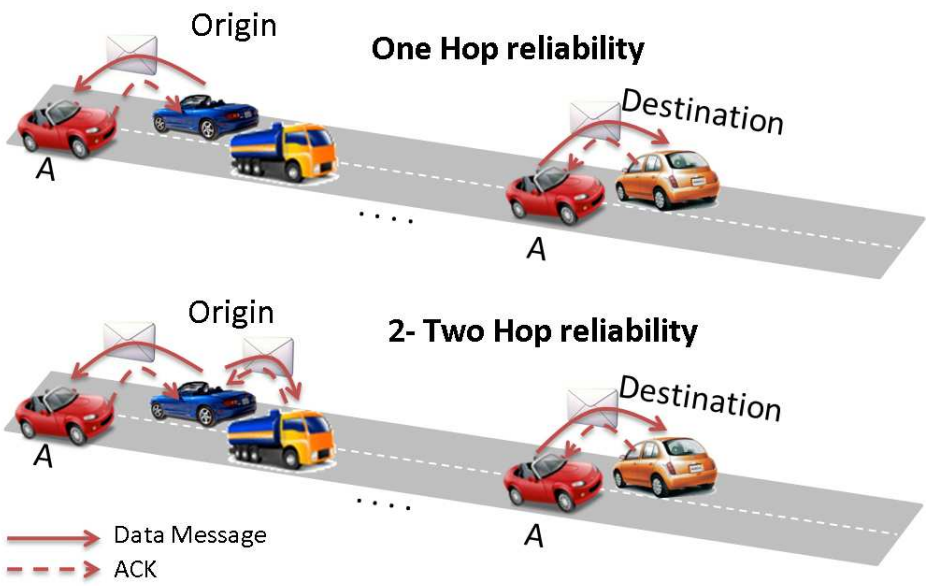

Figure 2.3: One hop reliability and two hop reliability techniques

n copies of it. The message tends to propagate through the network in a controlled flooding that resembles a three.

Epidemic routing (Vahdat, & Becker, 2000), consists of the spreading of the message as it occurs the spreading of an virus in an epidemic situation. Each node that receives the message rebroadcasts it to every other node it encounters. The contaminated nodes just keep one copy of the message. This approach is extremely effective, but presents a high resource consumption rate.

Ramanathan, Hansen, Basu, Rosales-Hain & Krishnan (2007) present a prioritized version of the epidemic routing. This technique imposes a partial ordering on the messages based on costs to destination, source, and expiry time. The costs are derived from the link availability information. The technique successfully maintains

a gradient of replication density that decreases with increasing distance from the destination. Even though it is also a resource intensive technique it presents lower costs and higher delivery rates than simple epidemic routing.

Forwarding based strategies

Location based routing (Jones, 2006) techniques use geographical information, such as Global Positioning System (GPS) data, to forward the data. This strategy is the forwarding one that demands the smallest amount of knowledge of the network structure. With the position information they can estimate the costs and direction to forward the messages.

Source routing, strategies calculate the whole path in the origin, prior than sending the packet. This kind of strategy needs to have a fairly consistent view of the network to work properly. On the other hand, on per-hop routing (Jones, 2006), the decisions of which path to take are done in a hop per hop basis when the message arrives at each hop. Instead of computing the next hop for each message the per-contact routing technique recomputed its routing table each time a contact arrives and its knowledge of the network increases.

Instead of routing with global contact knowledge, Liu & Wu (2007) propose a simplified DTN model and a hierarchical routing algorithm which routes on contact information with three combined methods.

DTN is a young and expanding field. VDTN has huge potential because of the imminent appearance of vehicular devices capable of wireless communications. These will operate on a very demanding environment, with intermittent connectivity, where an end-to-end path may not always be found. Even though routing and data dissemination have been the focus of research, areas such as security, topology management, transport layer issues, and higher protocol level concerns are equally important. They present problems that will need to be addressed in the near future. DTN and in particular VDTN is an attractive research field exactly for the reason that to achieve the envisioned future of ubiquitous connectivity, we need a solution for these open problems.

2.4 Data Transmission

2.4.1 Wireless link Protocol Effectiveness

2.4.2 WiFi

2.4.3 Delay Tolerant Data Dissemination

Data dissemination refers to data-centric communications protocols. The Data Mule project (Shah, Roy, Jain & Brunette, 2003) and the Message Ferrying scheme (Tariq, Ammar, Zegura, 2006), are two of the most well known data dissemination

algorithms for DTNs. They were designed for sensor networks. They propose the use of mobile nodes to collect data from the sensors, buffer it, and deliver the collected data to a sink. The MULEs (Mobile Ubiquitous LAN Extensions) and ferries utilize nodes navigating through the sensor network to collect data in 'mobile caches'. According to the Data Mule project, all the nodes are fixed and only the cache is mobile. Message Ferrying (Tariq, Ammar, Zegura, 2006) also considers mobile nodes, but in this approach the nodes are required to follow specific paths and even move in order to help message delivering.

The SPAWN protocol introduced by Das, Nandan, Pau, Sanadidi, & Gerla, (2004) and Nandan, Das, Pau, Sanadidi, & Gerla (2005) discusses how vehicles should interact to accommodate swarming protocols, such as BitTorrent traffic. In SPAWN, the nodes passing through Access Points (APs) collect data that they subsequently exchange among nearby nodes. Nodes are often required to carry traffic useless to them and the BitTorrent protocol is bandwidth intensive, however, swarming protocols is an interesting and effective way for message dissemination among nodes in VDTNs.

Frangiadakis, Cmara, Filali, Loureiro & Roussopoulos (2008) propose a simple and efficient dissemination algorithm called Virtual Access Points (VAPs). This work focuses the problem of data dissemination in Infrastructure-to-Vehicle (I2V) manner. They are interested on extending the I2V network to areas where regular

access points are not deployed. When a vehicle moves near an Access Point (AP), and receives a message, this vehicle becomes responsible for re-broadcasting it over the uncovered areas. Thus, it helps to spread the messages through the network, and the mobile nodes act as Virtual Access Points for nodes on thee the regions that do not have a real AP. This behavior is exemplified in Error: Reference source not found, where node A receives a message from the AP and afterward rebroadcasts it in a non covered area. For all practical purposes there is no difference between the services provided by the AP and the VAPs. The propagation mechanism is cooperative, and a node only acts as a VAP if it is outside any AP coverage area.

Transport issues

The greatest part of the research for DTNs has focused on routing and data dissemination algorithms. However, many other aspects present interesting and valuable challenges. The transport layer is certainly one of the layers that need special attention. Most of the services offered by existing transport layer protocols, such as TCP, have been ignored. For example, end-to-end connections, sequencing, congestion control and reliability are some of the most important features of the TCP protocol. Some of these services may be easily implemented in DTNs while others will require a fairly amount of future research. We will focus here reliability approaches to ensure message delivery on the DTNs.

Hop-by-hop reliability (Fall, 2003) is the most basic and simple reliability

strategies to ensure data delivery on DTNs. Each time a node receives a message, it sends an acknowledgement (ACK) of it reception and after that assumes the responsibility for this message across a defined region. For this case an end-to-end ACK is not possible, unless it is a completely new message generated by the destination.

The lack of end-to-end reliability of the hop-by-hop approach may be a problem for a series of applications. One of the ways to overcome this problem is the use of Active Receipt (Harras, & Almeroth, 2006). Active receipt is basically an end-to-end acknowledgment created by the destination, addressed to the source of the original message. The receipt is actively sent back through the network. In truth it is a new message that is propagated through the network. Active receipts solve the problem of end-to-end reliability but the price to pay for it may be too high in some situations. Passive Receipt (Harras, & Almeroth, 2006), is another method created to provide end-to-end reliability with a lower cost. The high price of the Active Receipt comes from the generation of two messages on the network instead of just one. To use the same term of epidemic routing, now we do have two messages infecting nodes instead of just one. In this case what Passive Receipt introduces is exactly the concept of an implicit receipt, instead of an active one. The destination, instead of creating a new active receipt, it creates an implicit kill message for the first one. The kill message works as a cure for the infected nodes, when they receive

this message they know the message arrived to the destination and that they do not need to rebroadcast the original message. The message is rebroadcast only if the cured nodes met other node that is re-broadcasting the original message. The flux of message is lower than the one generated by the active receipt, and the end-to-end reliability is guaranteed, since eventually the source will also receive the passive receipt.

An interesting solution for the end-to-end reliability is also proposed by Harras, & Almeroth (2006) and takes advantage of the number of multiple network infrastructures available nowadays. On the Network-Bridged Receipt approach the nodes may use a different medium access mode to delivery ACKs. For example, while the cell phone network may not present the required data rate for a specific application, or even present a high cost. The cell network may present more than reasonable bandwidth at a cost effective to send small ACK messages.

Delivery schemes

Direct transmission and flooding (Wang, & Wu, 2006) are two of the most simple delivery schemes possible. On direct transmission a node simply transmits the message direct to the destination. On flooding schemes the nodes transmit the message to all other nodes it may encounter. The analysis of both schemes is simple since the node behavior is straightforward to predict.

Epidemic dissemination schemes are also extremely popular for VDTNs. For example, the Shared Wireless Infostation Model (SWIM) presents an epidemic Markov dissemination scheme, (Small & Haas, 2003). The scheme is further analyzed and refined in (Small & Haas, 2005). Wang, Dang, & Wu (2007) present a more diverse description of dissemination models.

Queue Management

The way nodes manage their queues is also a determinant component in the de performance of algorithms for VDTNs. The way one model the queues determines, among others, the way nodes will discard old messages and this in consequence will, possibly, affect the network delivery ratio. The generic queuing analytic framework is introduced by Wang & Wu (2007) is a good start point for a simple queue model for VDTNs. The models described by Wang & Wu (2007) are infinite and finite buffer space. For the infinite buffer space the node's queue is considered to have infinite length. For the finite buffer space it is assumed that each node may hold at most k messages on the queue.

Niyato,, Wang, & Teo (2009) present an analytical queuing model based on discrete time Markov chains. This work also proposes models for queue performance measures for VDTNs. The proposed performance measures are: Average Number of Packets in Queue of a Mobile Router, Throughput and Average Packet Delivery

Delay

Applications

Research on VDTNs in the last few years has focused, among other topics, on using VDTNs in road safety applications. Research such as Xu, Mark, Ko & Sengupta (2004) evaluate the feasibility of using dedicated short range communication to warn vehicles about road accidents. Yang, Liu, Zhao, & Vaidya (2004) propose the use of V2V to warn vehicles about road conditions and demonstrate the potential of DTNs for real life applications.

2.4.4 Data dissemination

Data dissemination in a system can be push-based when the data are repeatedly broadcasted in a broadcast cycle, or pull based, when data is explicitly requested by the users, or hybrid using mixed push-pull approach. Push or push-pull based data delivery is very attractive in environments like the ones considered and has been studied extensively in various environments [AFZ97, AAFZ95, AFZ97, FZ98, SRB97, KM04a, ST00].

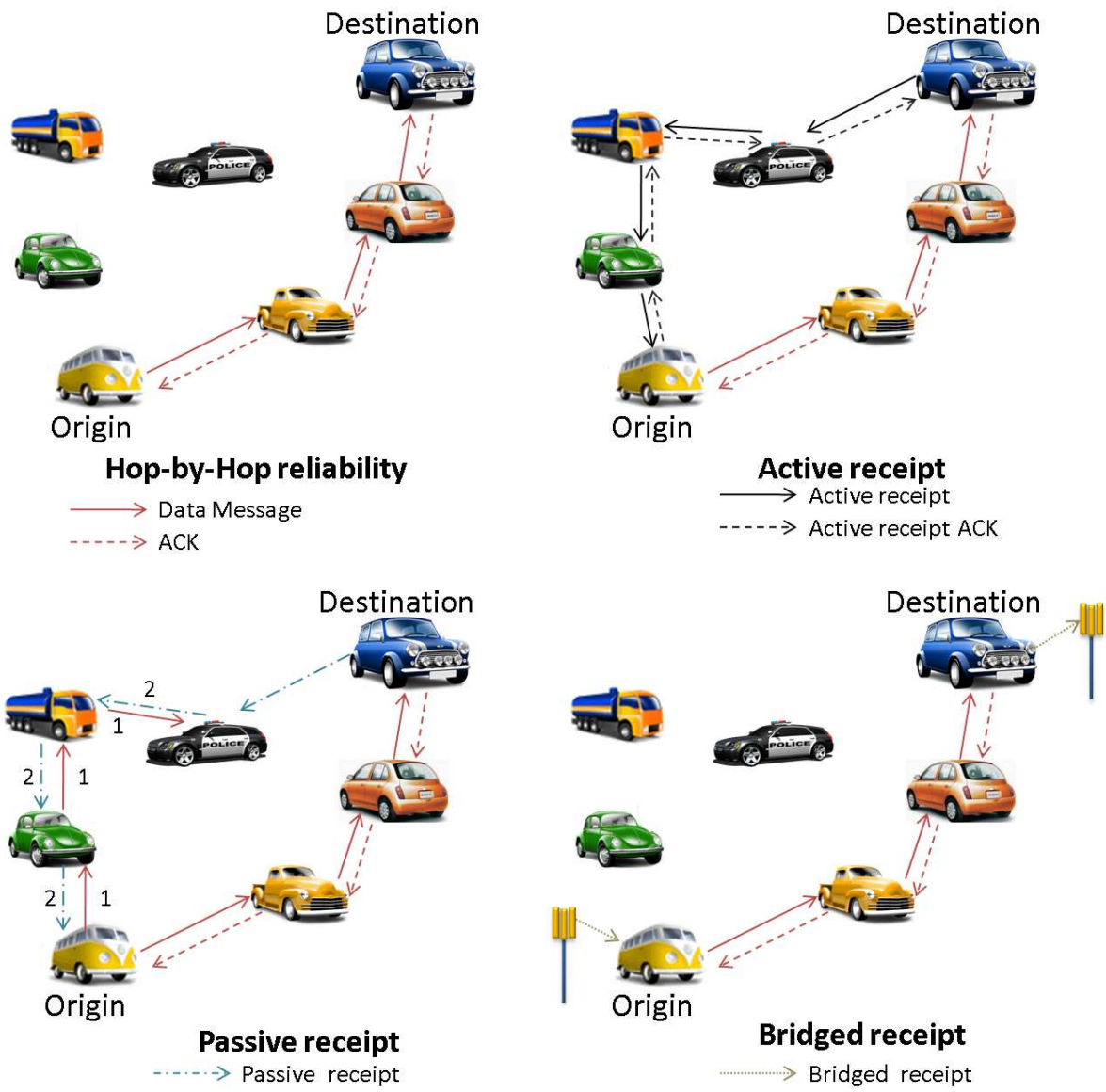

Figure 2.4: Comparison among the reliability approaches messages

2.5 User Mobility

2.5.1 Traces

For the simulations, real traces were used whenever available. Real traces are often used in simulations in related work for measuring these systems efficiency and some of them are published. CRAWDAD[KHA], provides a large collection of user traces. However, the applicability of using a set of traces can depend on the trace characteristics and the characteristics of the system we are trying to analyze.

To simulate the Mobile Cache system, we needed a large realistic mobility trace. We considered using a real trace from CRAWDAD[KHA], but unfortunately we needed a trace of ubiquitous, always on devices and a large enough trace of mobile users is not yet available. [JHP+03] only traces public transportation vehicles and paths. We used the samples developed from LST by Naumov et al. instead[NBG06]. The trace used is a large (compressed 668M, ns-2 movement file format) file that we had to break into parts to simulate efficiently. Another reason we chose to examine this trace is to find out in which aspects RMPG [KFHM04] can use the traces to create trips and day to day patterns for vehicles. [HFBF06] by Harri et al. that is based on CANU Mobility Simulation Environment can support ns-2 and so repeating the simulation with traces produced by it should be a straightforward and interesting future work.

2.5.2 User mobility and data transmission

The effect of mobility is discussed in recent papers [FVDN04], but both the analysis and solutions proposed include deadlines. Note that this is a pull-based system and analysis. Furthermore, it supposes that the system knows each user's available time a priori. In a real life and highly mobile environment this will be unknown and difficult to predict since it will depend on the user's future path, on possible interference, signal reflections, the movement of other "obstacles" in the area, and other conditions.

Mobility model

User mobility directly influences the network structure. The way nodes move, or do not move, may influence: the retransmission delays, frequency of contacts among nodes and energy decay. Mobility can either provide the chance for new high quality contact or well the break stable links already established. Different mobility model

Apart from static placement of nodes, probably the simplest mobility model is the Random Walk-based model (Zonoozi, & Dassanayake, 1997). On this mobility model nodes choose random points on the space and move towards these points with random speeds. The three basic steps for a random way point algorithm, as described by Bettstetter, Hartenstein, & Perez-Costa (2002) are: first the

node chooses randomly a destination, after that the node goes in direction to that destination with a random speed, the third steps is wait for a random period of time at the destination point. Some minor variants of this process are also possible, for example Spyropoulos, Psounis, Raghavendra (2006) consider random directions instead of positions, but in the end the main concept is the same.

Some well used techniques use different distributions to movement the nodes. The main advantages of using these distribution based mobility model are that, not only the mathematical model of a well known distribution is easy to implement, but also it is easy to analyze the network behavior afterwards. For example, knowing the nodes distribution is easy to calculate the probability of a node cross a specific network area. Some well used distributions are: Normal, poison and exponential.

Markovian mobility models are also a popular choice to model the mobility movement. The main goal of using Markov chains is to create more realistic movement models (Chiang, 1998) (Campos, Otero, & Moraes, 2004) with real drivers actions, such the movement in the same direction and in adjacent directions, acceleration, stops and sharp turns ; it can also simulate some special actions, such as acceleration/deceleration, sharp turns, stops and sudden stops.

Other model designed to provide realistic mobility patterns, introduced by Haerri, Bonnet, Filali, (2007), is Kinetic Graphs. This method tries to capture the dynamics of mobile structures and accordingly develop an efficient maintenance for

them. Unlike static graphs, kinetic graphs are assumed to be continuously changing and edges are represented by time-varying weights. Kinetic graphs are a natural extension for static graphs and provide solutions to similar problems, such as convex hulls, spanning trees or connected dominating sets, but for continuously mobile networks. This mobility model is implemented in a tool called VanetMobiSim, (Fiore, Haerri, Bonnet, Filali, 2007), that can generate realistic mobility patterns.

2.5.3 Modeling Techniques for VDTNs

Analytical studies perform an important role in the evaluation and, in consequence, in the development of protocols in every area, for vehicular delay tolerant networks it is not different. However, the constraints of DTNs are somehow particular, compared to traditional wired and wireless networks, the same analytical models and constraints may not hold for DTN environment. An analytical model, or study, "is a proven approach for studying system performance, revealing underlying characteristics, and evaluating communication protocols" (Wang, Dang, & Wu, 2007). Theoretical works, like the one of Niyato, Wang, & Teo (2009), provide the indication and comparison basis for other simulation or test-beds experiments.

Many factors may influence the analytical results of an experiment, e.g. node density, capacity, physical and medium access control characteristics. However the three main factors are: mobility model, data delivery scheme and queue management

(Wang, Dang, & Wu, 2007).

Chapter 3

Effects of user mobility - Mobile Cache

3.1 Introduction

In the near future ubiquitous communication devices and large city areas will be wifi-enabled. This will create a rich environment with new applications and services and enhance existing ones. Many of these new services will be data-centric in the sense that the user will be interested to obtain some data or learn some information (e.g. speed, position, acceleration/deceleration) of the nearest users and not communicate with some specific user based on his ID. For other services, the data will be highly correlated with the current and future location of the user (e.g. directions to a location, local advertising offers, nearest stores, nearest parking spaces etc.). As a result, needed data will frequently be common among users close to each other. Therefore, data broadcasting will consist an effective and scalable solution, especially since the wifi environment is inherently locally broadcast.

Wifi as well as WiMax can efficiently support broadcasting. Furthermore, future protocols for mobile communications should be expected to efficiently support

broadcasting. As an example, consider cooperative collision avoidance. The Federal Communication Commission (FCC) has assigned a spectrum of 75 MHz in the 5.9GHz to facilitate both public safety and private operations in roadside to vehicle and vehicle to vehicle operations [XRDD03], which is expected to lead to widespread communication infrastructure in vehicles using broadcast methods. At the same time, wide metropolitan areas will be wifi enabled. Already a number of cities have plans to cover most of their area or large parts with hot-spots. This will create island of connectivity. Google plans to allow free access to wifi hot-spots in San Francisco. It is also possible that in the near future users will decide to join private hot-spots enhancing each other communication capabilities [EP05]. Of course, the available bandwidth will be limited and further reduced by connectivity issues. Bychkovsy et. all[BHM+06] measured link layer connectivity at vehicular speeds to be 13 seconds, the median connection upload bandwidth 30 KBytes/s, and the mean duration between successful associations to APs is 75 seconds. In the case where more vehicles use wireless connections the available bandwidth might be less and it might be used for more than one application. Therefore we need to use it effectively. In scenarios where users are pedestrians, intermittent connectivity should also be expected, especially in an indoor environment, or when the radio of the user has smaller radius due to its position or lower power level.

3.2 Mobile-Cache: Intuition

We argue that existing analysis on caching becomes irrelevant for highly mobile environments. Here the server needs to deliver the requested object before the user moves out of range. The utility from server responses abruptly drops to zero for out of range users. As a result, what defines the efficiency of the system is the rate of objects successfully delivered rather than small delays in delivery.

Note also that a purely push approach does not make much sense since the user will stay connected for a very small fraction of the time needed to cycle even once through the database. Therefore, we consider the following systems: (i)**Unicast**, a straightforward albeit suboptimal approach that is implemented in some real deployments, and is used in this chapter as a measure for comparison. (ii)**Multicast**, that is pull based but takes advantage of the fact that multiple vehicles may ask for the same data. Using this approach, the BS responds to queries for trip segments using a "well known" 802.11 multicast address. The response data will be "pushed" to all other users in range as well, possibly answering additional queries. (iii)**Mobile-Cache**, where each BS acts in two phases, first collecting queries, and then answering using multicast. However it also proactively caches segment data most likely to be queried in a program that is multicast along with the other query answers. An index is used to inform vehicles not to ask a query for forthcoming segments. This greatly reduces the needed bandwidth. According to the analysis,

the optimal pull-push ratio depends on query frequencies, and on the user's mobility. Our algorithm captures frequencies and dynamically varies the cache size, that is the percentage of bandwidth used for program and index transmission.

Mobile-Cache dynamically adapts the bandwidth between push and pull according to both frequencies and the user mobility. In mobile environments, even for constant frequencies, the optimal ratio will depend on the user mobility, that will vary.

3.3 Analysis

We are concerned with a scenario where users are mobile. This means that depending on the speed of a vehicle, there is a T_{max} period of time available to answer a query. After that, the user will be out of range and the answer is considered lost. For this reason, the effectiveness of a method will be measured by the percentage of successfully completed queries and not query answer times.

Let v be the velocity of a vehicle, d the distance the vehicle will be in range ($T_{max} < \frac{d}{v}$), bw the total available bandwidth, bw_c the total available bandwidth after collisions are considered, sz_q the size of a query, and sz_r be the size of a response.

Unicast: The max number of requests that can be successfully answered in

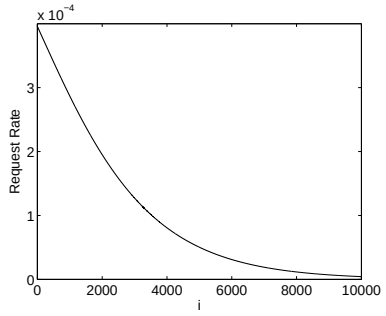

Figure 3.1: Example object request rates

this case will be limited by:

$$E(\frac{\text{\# answered requests}}{\text{sec}}) = \frac{bw_c}{sz_q + sz_r}$$

Multicast: The idea here is that since the medium is broadcast, the answers to any query will be received from all other vehicles at no extra cost. In fact, some of the requests are repeated more often. In the application used for simulation this happens when vehicles using the same road often are following similar paths and are likely to send request concerning the same road segments. In general, in most applications there are items that will be "hot" and others that will be requested much less often as shown in Figure 3.1.

Let $\mathrm{P}(x = i) = \lambda_i$ Probability Mass Function with λ_i the relative frequency for requesting item $i \leq \mathrm{N_{items}}$. Figure 3.1 depicts such a one.

Let N_{total} be the average number of queries over time from vehicles in range (e.g., how $\mathrm{E}(\frac{\#\text{requests}}{\text{sec}})$). N_{total} will depend on the number of vehicles in range at each

time and number of queries per vehicle. The first depends on the speed and density of the cars and for the purpose of our analysis we take it to be $(\frac{const_1}{v} + const_2)$, where $const_x$ constants. Note that in the simulation, as well as in real life, the density of the cars is not constant. In fact it will vary with speed, number of lanes, conditions and other factors. Therefore, in our analysis we focus on the efficiency of the methods at any given $\frac{N_{total}}{sec}$ rate.

$N_{total} = (\frac{const_1}{v} + const_2)const_3 = \frac{const_a}{v} + const_b$

Similarly to the unicast case, the maximum number of requests that can be multicasted over time is on the limit $N_{req} = \frac{bw_c}{sz_q + sz_r}$

However we must also all identical requests of other users that were answered because answers are multicasted.

Let req_i, $i \in \{1..N_{total}\}$ be the N_{total} total requests over a sec.

In fact, the average number of answered requests in the same period will be:

$$E(\frac{\text{\# answered requests}}{\text{sec}}) =$$

$$N_{req} + (N_{total} - N_{req}) \sum_{k=1}^{\text{N}_{\text{items}}} \left(\lambda_k(1 - (1 - \lambda_k)^{N_{list}})\right)$$

where N_{list} is number of broadband requests each vehicle can listen to while in range, $N_{list} = \frac{d}{v} \cdot \frac{bw_c}{sz_q + sz_r} - 1$

Proof: Over time period of a sec, there is enough time for N_{req} of the N_{total} requests, and so N_{req} will be answered directly, while the remaining $(N_{total} - N_{req})$ may be answered if they are identical to any of the multicasted messages. Each user will be able to listen to $N_{list} = \frac{d}{v} \cdot \frac{bw_p}{sz_q + sz_r} - 1$ multicasted messages per average. Let A the subset of Answered questions.

The probability that a random query req will match one of these messages $\{req_j, 0 \leq j \leq N_{list}\}$ is given by:

$$P\left(req \in A\right) = P\left(_{req_i = req_1}\right) + P\left(_{req_i \neq req_1, req_i = req_2}\right) + \ldots + P\left(_{req_i \neq req_1, \ldots, req_i = req_{N_{list}}}\right)$$

$$= \sum_{k=1}^{\mathrm{N_{items}}} \lambda_k^2 + \sum_{k=1}^{\mathrm{N_{items}}} \lambda_k^2 (1-\lambda_k) + \sum_{k=1}^{\mathrm{N_{items}}} \lambda_k^2 (1-\lambda_k)^2 + \ldots + \sum_{k=1}^{\mathrm{N_{items}}} \lambda_k^2 (1-\lambda_k)^{N_{list}}, \text{ and so}$$

$$P\left(req \in A\right) =$$

$$\sum_{k=1}^{\mathrm{N_{items}}} \left(p_k^2 + p_k^2(1-p_k) + p_k^2(1-p_k)^2 + \ldots + p_k^2(1-p_k)^{N_{list}}\right)$$

$$P\left(req \in A\right) = \sum_{k=1}^{\mathrm{N_{items}}} \left(p_k(1 - (1-p_k)^{N_{list}}\right) = P_{MC}$$

So for N_{req} queries it is E($req \in |A|$)=1,

while for each of the remaining E($req \in |A|$)=P_{MC}

Without loss of generality assume the N_{req} directly answered queries were the $\{req_i, i \leq N_{req}\}$ ones (e.g. the first to be requested)

Then by linearity of expectations:

$\mathrm{E}(|A|)=\mathrm{E}(req_1 \in A)+...+\mathrm{E}(req_{N_{req}} \in A)+...+\mathrm{E}(req_{N_{total}} \in A)$

$\mathrm{E}(|A|)=N_{req} + (N_{total} - N_{req}) \sum_{k=1}^{\mathrm{N_{items}}} \left(\lambda_k(1-(1-\lambda_k)^{N_{list}})\right)$

Mobile-Cache: First we analyze the case when all available bandwidth is used to send the cached program. Suppose the server has perfect knowledge of request frequencies without any requests being transmitted. Let bw_c the total available bandwidth after collisions are considered. The number of responses each vehicle can listen to will now be:

$$\mathrm{N}'_{list} = \frac{d}{v} \cdot \frac{bw_c}{sz_r}$$

Let $\Lambda_i = \sum_{k=1}^{i} \lambda_k$, and N_{total} be the total number of requests.
Then the expected number of answered requests will be (linearity of expectations):

$$E(\tfrac{\text{\# answered requests}}{\text{sec}}) = \mathrm{N}_{total} \cdot \Lambda_{\mathrm{N}'_l}$$

As we can see in the figure 3.2 , using a cached outperforms multicast and both are significantly more efficient than unicast. If we assume 30 KB/sec with 13 sec of connectivity as in [BHM+06], that is shown as 390 KB/sec for 1 sec

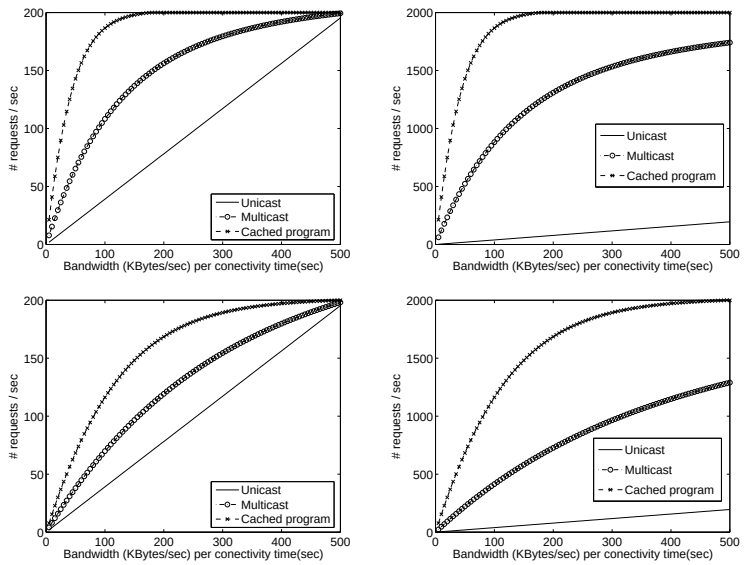

Figure 3.2: a) speed: 70mph, total queries/sec: 200, b) speed: 70mph, total queries/sec: 2000 c) speed: 25mph, total queries/sec: 200 d) speed: 25mph, total queries/sec: 2000

of connectivity. However, in our analysis we assumed that we know exactly the frequencies of the query distributions. In reality we will only have an estimate about the future request rate for each segment. Furthermore, these rates will vary dynamically. In the extreme case where we have no information, the average number of answered requests will be: $\mathrm{N}_{total} \cdot \frac{\frac{bw_c}{sz_r}}{\text{total } \# \text{ items}}$, which is inefficient. This means that a complete system should use something more flexible than a pure push approach.

Also, in the case where the traffic is low on a given area, using multicast could potentially lead to answering all the request with far less messages, causing much less congestion a channel possibly used for other applications and less interference to neighboring APs. Finally, if we don't allow for multicast, some user queries may

never be answered, which is not fair.

We solve these problems with Mobile-Cache. Mobile-Cache uses part of the available bandwidth for the cached program and part for multicast. Time is divided to periods, and after each period the bandwidth ratio used for cache is adjusted and a new program is decided.

Multicast is used to estimate the frequency of request for each segment as well as answer infrequent questions. We keep track of request rates for each segment and use them to compute a "temperature" for each segment. After every period, the temperature of each object is its previous temperature times an aging factor of .9 plus the new number of requests for that object. The "hottest" segments will form the new program.

Furthermore, each time a user contacts the BS with a set of segment requests, it includes in the first of these requests the ratio of its segments that were in the program over the total trip size. This is gives us an indication of the efficiency of our cache e_{obs} =E(ratio). At the beginning of each turn, we compute the estimated cache efficiency in accordance to section 3.3 as the sum of the temperatures of all segments in the program over the total sum $e_{est} = \frac{\sum_{program} temp}{\sum_{total} temp}$. This gives us a target cache efficiency that would be reached if temperature information is accurate. Whenever the measured cache efficiency is greater than a percentage of the target, which we set to be .9, this signifies that the bandwidth used for multicast is enough

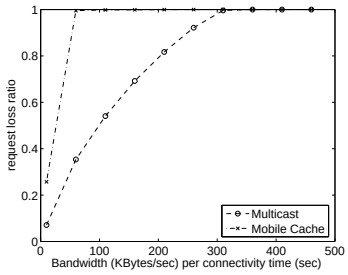

Figure 3.3: Mobile-Cache Performance: Loss ratio as a function of the available bandwidth per second of connectivity

to inform us about the frequency of segment requests. Mobile-Cache computes $r_\delta = r_{old} + .9(e_{obs} - e_{est})$ where r_{old} the ratio of cache to multicast bandwidth.

Mobile-Cache also monitors sz_u the amount of space allocated to multicast not used. If $\frac{sz_u}{sz_r + sz_q} > 4$, the unused space corresponds to four or more requests and it computes $r_{reduced} = \frac{r_old \cdot bw_p - sz_r}{bw_p}$. In this case the new ratio will be $\min(r_\delta, r_{reduced})$. This allows us to reduce the size of cache when there are very few vehicles. No messages will be lost since there is enough bandwidth for multicast.

Finally, we never allow r to be more than .8. This is to be able to respond to changes, as well as to allow infrequent queries to be answered.

3.4 Simulation

We use publicly available traces realistic VANET traces to Simulate the Mobile-Cache system. Our simulator can take any ns-2 movement file as input. The one we experimented with is an extensive simulation with a large movement

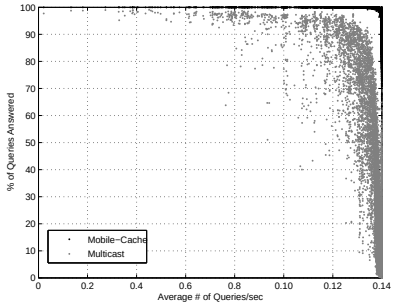

Figure 3.4: Mobile-Cache vs. Multicast: Avg. % of Q vs. rate of Q/sec

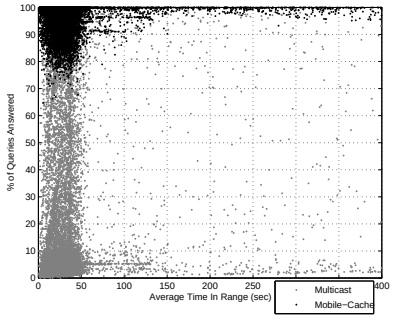

Figure 3.5: Mobile-Cache vs. Multicast: Avg. % of Q vs. available time in range

file (compressed 668M, ns-2 movement file format).We use a two stage simulation. We first run the mobility simulation and note all events of users entering or leaving an AP area. We then presort this list for each AP and use it to introduce these event efficiently. In the following figures each point represents the average statistics collected from a unique simulated Access Point. This allows us to realistically simulate Mobile-Cache.

Note that in the Figures, black points represent Mobile-Cache, and grey points multicast. Each point represents average performance for an AP Therefore there is

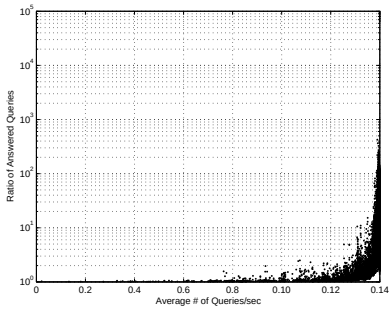

Figure 3.6: Mobile-Cache Performance, log scale: Ratio of Q vs. average Q rate

an equal number of black and grey points in each figure. As the speed and mobility pattern of cars is the one provided by the mobility trace, we are not free to vary it but just measure and report it.

Mobile-Cache outperforms the multicast even when we do not take into account the congestion during its larger pull phase. This is because it is designed with a mobile environment in mind. However, we used no additional functionality to decide what the users speeds and profiles are, or to predict how this will evolve in the future. Because of this we believe that the system's performance can be further improved. Mobile-Cache serves its use as an example that takes a first step to improve an already efficient method of data dissemination by taking into consideration the new parameters introduced by the mobile environment.

Figure 3.8 further supports the analysis showing the effect of mobility to performance. We see that there are two sets of users and that how performance varies largely depends on the case we examine. The APs lying along each axis are affected

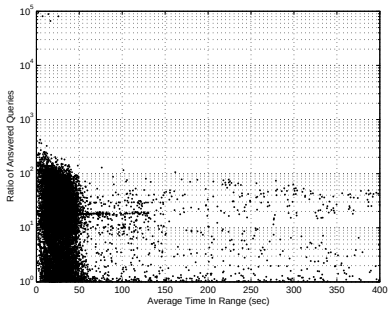

Figure 3.7: Mobile-Cache Performance, log scale: Ratio of Q vs. average time in range

differently when load and the mobility of users vary. Also note that these two sets of users are probably because in the trace there exist both fast, high volume roads and slower ones with less traffic. Of course a small part of the high volume roads are congested as well.

In conclusion, in a mobile environment reception of the data is more important than small delays and we are the first in our best knowledge to provide the analysis of push, pull and mixed push - pull ratio under this light of step utility. Mobile-Cache is built with this analysis in mind and uses a smart way to dynamically estimate frequencies and avoid overheads.

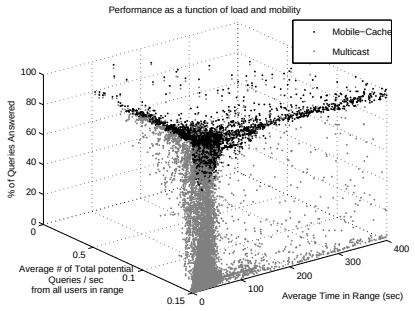

Figure 3.8: Mobile-Cache Performance: Parameter Space

Chapter 4

PEGASUS

4.1 Introduction

Wireless access technologies are widely deployed in today's world, and they are a primary means in providing Internet connectivity to mobile users. Connectivity can be improved from access to multiple mediums such as WiFi and cellular. In this chapter we focus on utilizing 802.11 networks as the principal communication technology.

WiFi networks are usually operated by private parties where wireless routers are self-contained limited-range segments. They offer high bit rates in comparison to mediums with longer reach and they are relative inexpensive to operate. The number of 802.11 networks has grown significantly in the last 5 years. According to recent studies [BHM+06] the number of home-deployed wireless routers in the US exceeds 15 million and rising. Such statistics suggest that many of these networks may overlap and allow mobile users to remain in range of some WLAN for continuous periods of time. The challenges of using them from a moving vehicle emerge largely

due to their independent nature and short range. Movement from area covered by one access point (AP) to an area covered by another access point often requires a user to acquire a new IP address, and reconstruct all of the connections that were broken because of the IP change. In addition, each WLAN usually operates on its own private subnet and NATs the internal network to the outside world. As a consequence, users have to adapt to this behavior, and many applications cannot handle breaks in connectivity.

The problems grow in magnitude and complexity when mobile users traveling at high velocities (i.e. by car) are considered. The average connection to a single WLAN for such client is only 6 to 15 seconds. Moreover, due to time spent for DHCP and other conventional connection setup procedures, the precious connectivity time is mostly wasted. Therefore, today, rapidly moving users cannot use WiFi and have to rely on other expensive and bandwidth-limited wireless access such as cellular.

PEGASUS:

To address the difficulties described above we have designed PEGASUS. PEGASUS is a system built to deal with rapid 802.11 access point connection switches; however it can also use cellular or other long range mediums when WiFi is not available in the area. We focus on utilizing higher bandwidth connections and we abstract the underlying network management specifics. PEGASUS transparently switches between 802.11 access points, or even different mediums (WiFi or Cellular)

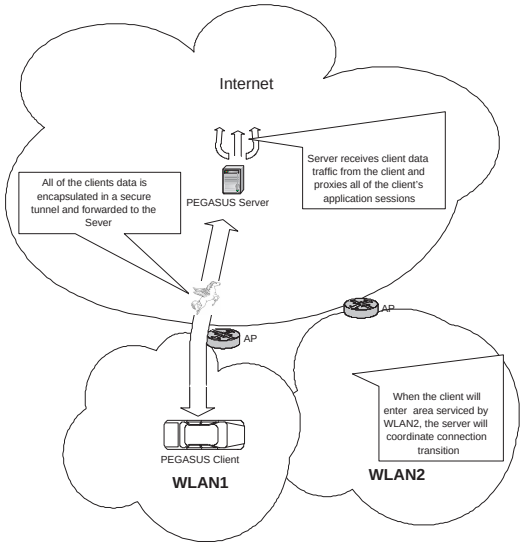

Figure 4.1: PEGASUS - High Level Overview

presenting a constant IP and persistent connectivity appearance to users. To support transparency as clients move from one WLAN to another we use PEGASUS Server (PegSvc) to manage all connections. In addition, PegSvc hosts a connection database to aid clients as they move.

Our system's efficiency is based on re-using connection knowledge among many clients and pre-fetching connection candidates on the client path to minimize connection setup overheads. The overall PEGASUS architecture is designed to support very large networks with WAP security to protect both - system clients and network operators.

This research was inspired by ongoing work in the area of wireless connectivity for moving vehicles. Projects such as Drive-Thru Internet [OK04a] have illustrated

the feasibility of connecting to a WLAN on high speeds and effective use of its bandwidth. The CarTel [HBZ+06] project illustrated an approach that maps numerous WLANs on the client route, and uses that information for future client connections. Both projects offer a valuable insight in the vehicle WLAN connectivity, and share our view of reusing existing network protocols without requiring clients and applications to move to other transport layer approaches such as mobile IP. However, PEGASUS moves one step further, and offers a comprehensive solution to wireless connectivity through the existing infrastructure.

The above mentioned projects treat WLAN networks as separate domains and concentrate on solutions that deal with changing IP addresses and intermittent connectivity local to the client. In addition, both of them deal exclusively with the wireless network mediums. These works recognized a need to minimize the connection setup period when the client roams. The client software caches DHCP leases and reuses them when the client is back in range of a previously known access point. In PEGASUS, however, to achieve efficiency we propose to reuse a DHCP cache globally. PEGASUS caches all of the DCHP connections from all of the clients in a global cache, and continuously reuses them. Since DHCP is bound to the client MAC address, in order to reuse DHCP connections, users change their MAC address to a value handed-in by the PegSvc. Once a user moves on to the next connection, it changes the MAC address again. This concept of recycling acquired

DHCP connections and using a different DHCP identity at each independent access point island is the core concept that allows PEGASUS to achieve its efficiency and scalability.

The DHCP cache on PegSvc is built dynamically by clients as they discover new access points and connect to them. Any new connection created by a client will also be available to other clients passing through that area after the first. PegSvc builds up a global database of access point layout and DHCP connection and treats these as common infrastructure resources that are reused by many clients. Rapid switch between cached DHCP connections is achieved by reducing connection setup time. The larger the global cache, the higher the PEGASUS service effectiveness. To speed up the cache built up, Access Point owners can participate by creating connections to their access points using a PEGASUS utility. In this case APs can be secure with WAP and still, clients will be able to connect to them when in range. To the best of our knowledge PEGASUS is the first system to allow usage of secured access points for vehicular communication.

To coordinate client and manager applications, PEGASUS employs a management protocol to maximize connection time utilization for useful data transfer and to switch to the next access point on the path before the connection deterioration. Figure 4.1 presents a high level overview of PEGASUS. The mobile client in the automobile is connected to WLAN1 network. Client applications use the wireless

connection for Internet, and all application sessions are encapsulated in a tunnel and sent to the PEGASUS server. Before the vehicle leaves the area serviced by WLAN1, the server will send next connection information, coordinating the client's switch to WLAN2.

We assume that every WLAN is independently managed, so we deal with different ISPs, private address spaces and NATs. As depicted on Figure 4.1, to handle such heterogeneity, client's traffic is tunneled to the PEGASUS server. The server can be operated by a third party and acts as a multiplex point for all client Internet communications. PegSvc attempts to predict the client movement through deployed WLANs and offers choices for the next access point connection. The switch from one AP to another will not sever the ongoing client application sessions. Moreover, since the server acts as fixed peer to the non-mobile connection endpoints, it buffers network packets to smooth possible connectivity dead spots. All of the tunneled traffic is encrypted to offer extra security for the client data.

In summary PEGASUS strives to maintain a seamless, high throughput TCP connection during handovers. For efficiency we reuse a global DHCP connection cache among all clients and attempt to predict connection candidates on the client's path. The PEGASUS connection switch is transparent to client applications, and does not impose modifications neither to the infrastructure of deployed networks nor the Internet Protocol stack. We allow participation of secured access point and

we offer client and network operator security with authentication and encryption services. Finally, PEGASUS is not limited to WiFi and will use any wireless medium to sustain client connectivity.

The rest of this chapter is structured as follows: Section 4.2 describes PEGASUS, and explains the reasoning behind our approach. Section 4.3 presents measurements and results from the study with prototype implementations, and Section 4.4 concludes this work and presents future research directions.

4.2 Architecture

The main objective of PEGASUS is to provide a solution that will present client applications with an appearance of a consistent connection, optimize utilization of individual connection, and minimize the connection transfer overheads. In addition we want to be able to support a large client base, offer better security, and create a system that is easy to deploy. In this section we present architecture for PEGASUS. First, we outline our assumptions about the underlying infrastructure available today. Next, we discuss the overall system architecture and introduce individual components and their responsibilities. Finally, we present the control protocol messaging interface and discuss the applicability of our approach.

4.2.1 Assumptions

- Availability of WLANs - with continuing deployment of wireless access point in the US households, and in accordance with reports from pervious research projects, we assume that our clients will travel in a more or less connected grid of WLAN connection spots, and they will be able to find an available WLAN network most of the time. Since PEGASUS operates on either open and secure APs with the consent of the access point owner, the assumption is realistic. The non-connectivity periods should be relatively brief, and PEGASUS can to fall back to non WiFi wireless network if need arises.

- Length of single WiFi connection - in 802.11b/g network connectivity can vary between access points, and AP range can span from 200m to 1000m or more. For 25% to 40% of the time as the client passes the "production zone" (area closer to the access point), the client will experience good connection quality. Once the client is ready to exit the "production zone," we would like to switch to the adjacent network for the next "production zone". Each access point connection can last from 5 seconds to almost a minute at various driving speeds [OK04a]. To accommodate frequent switches (every 15 to 20 seconds) we need to minimize connection setup overheads and avoid DHCP discoveries which can take up to 7 seconds each.

- No change to the underlying "in-situ" infrastructure - each WLAN is operated by a different provider, thus we have to accommodate switching to different IP addresses and private NAT domains, as well as using different security credentials for each access point. For example, each wireless access point today may use its own channel, SSID, and WEP key in secure networks. To have a realistic solution PEGASUS needs to use "in situ" infrastructure and avoid imposing additional hardware or network protocol requirements. Therefore, using something like Mobile IP or I-TCP is not possible.

- Utilization of multiple wireless mediums - mobile devices today, often have more then one type of a wireless interface. To deal with occasional intermittent connectivity of the mobile client when 802.11b/g wireless connection will not be accessible, PEGASUS will use other means to sustain connectivity.

- Network Security - - a complete solution needs means to protect user data, as well as ways to prevent network abuse and user illegal activities.

4.2.2 Requirements

The above mentioned assumptions were compiled into the following requirements list for PEGASUS:

- Transparent connectivity appearance to client applications (i.e Web, email

access, file transfer, etc. . .)

- Deployment on top of "in-situ" access points, without managing or changing the existing infrastructure, and support for any WLAN configurations
- Simple installation on clients and easy server deployment, without modifications to the existing operating systems and applications
- Support of the existing user equipment without requiring any custom or specialized hardware at the client or server devices
- Utilization of multiple network mediums available at the client
- Extensibility for future performance enhancements to further improve mobile connectivity
- Simple system deployment and dynamic growth of the system connection database
- Scale to support a growing number of clients
- Security for clients and network operators

4.2.3 System Architecture

In order to provide seamless connectivity in a mobile environment, and employ "in situ" network infrastructure, PEGASUS uses a service above the transport

layer for connectivity management, and masks the physical connection transitions by offering a virtual network interface with a constant IP address to the client applications. The primary idea of PEGASUS is to split the end-to-end connection to conceal the client IP address changes from the applications on the mobile end and fixed host services. The two main components that achieve the connection splitting are the client module that resides at the mobile node and the manager proxy that is located in the network. The client and the manager nodes communicate with each other via a control message protocol and hide connection transitions from the application layer sessions. Additionally, to survive the loss of connectivity for brief periods of time and still achieve persistent connectivity view, the manager and the client modules maintain connection states and offer session traffic buffering.

Figure 4.2 depicts an overview of our architecture. The client is composed of the following elements:

- Transparency Layer provides the user applications with an appearance of a constant IP address. The layer creates a virtual interface and modifies client routing to send all of the outgoing traffic via that interface

- Connection State Management Management resides just below the transparency layer and its primary function is optimization to handle volatile connection conditions of the mobile network. This layer offers client side buffering,

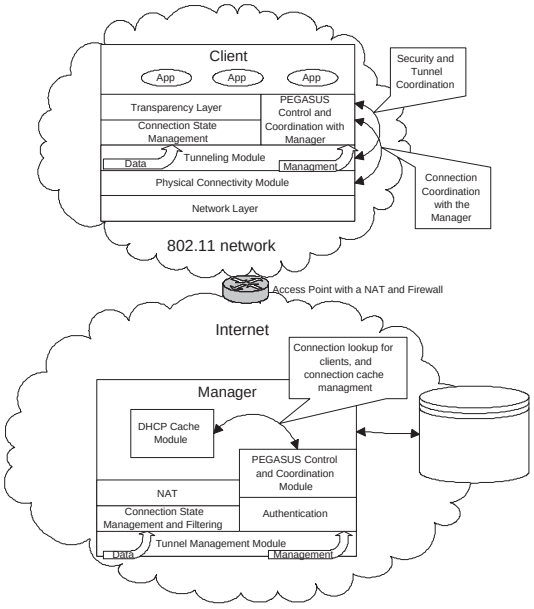

Figure 4.2: System Architecture

and it can track outgoing TCP connections to keep them alive during the moments of intermittent connectivity. To supplement the current functionality, this layer can be extended to notify applications that wish to have knowledge about actual current physical connectivity status.

- PEGASUS Control Module is responsible for communication with the manager to coordinate connection transfers and other manager supported services. This module tracks client movement, and keeps connection options received from the manager. In addition, this module offers an interface to authenticate with the server, and notify the server of new connections. All of the client-server control communication is handled by this module.

- Tunneling Module creates a UDP tunnel to the manager and forwards all of the traffic generated by the client applications and control module to the manger. This module supports open and encrypted tunnels. The tunnel parameters are negotiated during client authentication phase with the manager. Also, at this layer all of traffic received from the manager is classified as data and passed up to the Connection State Management module, or classified as management traffic and passed to the PEGASUS Control.

- Physical Connectivity Module is responsible for maintaining a physical connection at all times. This layer keeps track of the available connection mediums at the client, and monitors each medium for signal quality. The main focus in the current implementation is tracking of the 802.11 signal and detection of the signal deterioration. The module will ask PEGASUS control for a list of available connections which the manager has for client's location. The client scans for networks, and uses this list to select one with a good signal. In the cases, when the scan cannot match any of the found connections, the client will forward traffic over the secondary medium, and attempt to connect to 802.11 networks with DHCP. Once the connection is established, it is sent to the manager to add to the global cache. The purpose of using a connection from the manager list is to avoid DHCP discovery. The global cache contains connection information in the form of (MAC, IP, SSID, AuthInfo)

tuples. When clients use these tuples they take the identity of an already configured entity in the WLAN. Once they move, the cached identity can be reused for other transit users. Such connection information recycling allows PEGASUS to avoid setup overheads, and this scheme guarantees that we will only use a limited number of resources protecting the wireless network owner and his access point from abuse.

The complementary part of the clients in PEGASUS system is the manager proxy that multiplexes all of the client connections and stores them in a global DHCP cache to be recycled. The manager proxy server consists of the following elements:

- DHCP Cache Module stores all of the known connections created by PEGASUS clients. The cache expires old and stale information and updates the renewed connections. Along with the DHCP data the module keeps the connection locations. The location information is used to respond to client requests with access points on the client's path. The cache is dynamically populated as client nodes discover new access points, and by access point owners that want to participate in PEGASUS and create connections to their network. Every connection in the database can have set of filter rules to restrict client internet access. We do not promote such restrictions, but it allows access point owners to have more control over their network.

- PEGASUS Control Module responds to all incoming requests from the clients. The Control Module authenticates clients to use PEGASUS, and it responds to client connection requests with entries from DHCP Cache. In addition, the Control Module tracks client movement and connection usage, and updates client NAT entries to correctly route traffic when connection switches.

- NAT is a network address translation scheme used for connection splitting in the system. As the client moves from one connection to another, PEGASUS hides client mobility by NATing all of the client's connections. The client end points in the NAT are constantly updated to route client data to the correct connection.

- Connection State Management Module is the manager equivalent of the client Connection State Management Module. This piece is not finished; the full implementation would need to keep track of various protocols above IP to keep alive application sessions on the fixed host end, when the mobile nodes experience intermittent connectivity. The main purpose of Connection State Management module pair is to improve connection robustness in volatile mobile environments. The client side deals with the mobile end, and the server side handles the fixed host session end.

- Tunnel Management Module unpacks data and management traffic from the

client tunnels. The data traffic is forwarded to Connection State Management and NAT, while management traffic is forwarded to the PEGASUS Control. The tunnel security parameters are negotiated during client authentication.

4.2.4 Control Protocol Messaging

Our architecture requires client and manager proxy to maintain a persistent relationship for managing wireless connection transfers during client movement from an area serviced by one access point to the area serviced by the next access point. To achieve this we have developed a control protocol for PEGASUS clients and managers

Now, we present a quick overview messages that we support:

- authenticate - client sends this message when it connects to the PEGASUS to authenticate itself with the manager, and to negotiate tunnel encryption settings.

- connection_list_request - client sends this message to request a list of connections in its proximity. Manager will reply with a list of connections within 500 meters from the client. Furthermore, the client attempts to pre-fetch extra information to avoid delays when the connection deteriorates and a switch is desired.

- connection_in_use - client uses this message to notify server that uses this connection. The server can sometimes NACK, if the connection is already used by another client. When the connection is new, the client embeds the connection information in the message and server will add it to the cache. Also, if the server detects that the client moved on (by noticing a change in the client's tunnel end point), it will mark the connection for client's use without explicit "connection_in_use" message.

- connection_add - client uses this message to add a new connection to DHCP cache without actually using the connection for communication.

- ack/nack - used by the manager to allow/disallow client connection use.

This is the list of messages that PEGASUS currently supports; in the future, the protocol can be extended to support additional services and requirements.

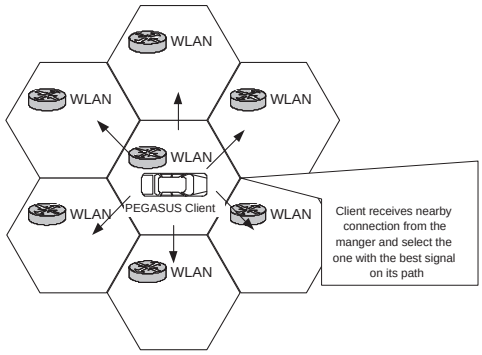

Figure 4.3: Client Connection Switch Options

To demonstrate control protocol usage Figure 4.3 depicts a simple use case. When a client needs a connection, it sends a "connection_request" to the manager, and receives a response with a list of connections in the proximity. With this information the mobile node can select a connection with the best signal, and it can predict the next one or two connections along its movement path. Once the client decides on the next connection, it sends a "connection_in_use" message, which the manager, can "ack" or "nack" depending on availability of that connection. In most scenarios, the manager will acknowledge the connection, and update the UDP tunnel and NAT mappings to route to a new client address. When client approaches the edge of the connectivity area, it will send another "connection_request" and transition to the next connection.

In cases when the client does not receive a connection from the manager it will try to find an open network and connect. Once connected, it will notify the manger about the new connection with "connection_in_use".

4.2.5 Applicability

The described above components comprise our approach. The client transparency layer achieves application connection transparency. The physical connection layer attempts to provide network connectivity at all times. The Tunneling layers at the client and server deal with private networks, NAT and firewalls at

802.11 access points. The tunnels are simple to establish and allow client traffic to remain transparent to the internal WLAN settings. Also, the tunneling provides the connection splitting mechanism between the rapidly moving client, and the fixed endpoints. Since all of the client connections are NATed at the manager - client mobility is hidden. The efficiency of the connection switching comes from the global DHCP cache, and pre-fetching of the connections on the client's path. Finally, PEGASUS connection management layers provide mechanisms to deal with brief periods of intermittent connectivity, and the system provides security with authentication and tunnel encryption.

The last system requirements that we stated was ease of deployment and scalability. At the client the required modification is a single executable module to abstract the physical connection. The manager proxies also run a module that inspects incoming traffic and runs NAT. PEGASUS proxies can be scaled by increasing a number of server machines and splitting the connection database among them by geographic regions. The proxies do not require any centralized communication or synchronization aside from client authentication services. As the number of clients in the system increase, one can install more managers and keep partitioning connection database on the basis of connection location. The overall infrastructure is very light and does not impose any additional rules on the deployed networks, and we hope current technology trends continue to introduce more mobile devices with

capabilities to connect to multiple wireless mediums, making them potential client devices in PEGASUS[RCC+04].

4.3 Measurements

PEGASUS is implemented on top of Ubuntu Linux distribution. To implement various routing and networking functionality on client and server we took "Click Modular Router" project [MKJK99] and extended it. In addition on the client, we have incorporated Wireless Extensions for Linux [Tou]. Thus PEGASUS will work with any 802.11 card supported by Linux.

To measure PEGASUS performance we have simulated a mobile environment in our lab. PEGASUS server is a Pentium III with Ubuntu Linux deployed in public domain. For "in situ" WLANs, we installed Linksys 54g access points that are available in any store configured with default factory settings. Every access point runs a firewall, NAT, and DHCP for its private network. Several of the APs are secure with WEP and we imported their connection information into PEGASUS DHCP cache manually. For the open access points the DHCP cache on the server is populated dynamically by clients that connect to every WLAN and send the DHCP connections to server. The client used for measurements is a regular laptop running Linux with ipw3945 Intel wireless card which is a standard for Dell laptops. To simulate movement, the client switches its wireless connection from one access point

to the next and the connectivity period to each WLAN depends on the simulated driving velocity.

To benchmark performance we use TTCP tests and a web browsing session. The TTCP application runs unaware of the ongoing physical connection transitions and measures end-to-end TCP bandwidth. In order to emulate different application behaviors we test with several TTCP configurations; we simulate large continuous data transfers, and multiple smaller data transfers. For the web browsing sessions we record response times and the number of times a web page comes back with status "400 Page Not Found".

To evaluate PEGASUS we developed several scenarios. First, we run our test suite without PEGASUS. The client uses a direct WiFi connection with no proxy involved to record a baseline performance metrics. Then, we run the tests with PegSvc proxing but without any connection transitions, the client remains connected to the same access point for the duration of the tests. Finally, we simulate several driving scenarios for velocities from 20 - 100 km/h (12 - 65 mph). First, we measure client performance when the client always has to request a fresh DHCP from every access point during connection transition. This is the worst case scenario for PEGASUS, since the client does not use global DHCP cache. Second, we measure performance when the client acquires DHCP from access points 50% of the time while the other 50% of the time it uses a connection from PEGASUS DHCP cache.

Finally, we evaluate performance of PEGASUS when the client uses a connection from PEGASUS DHCP cache for every transition, and does not need to do any DHCP requests.

For the simulations, we have assumed that a WLAN range is 250 meters in diameter, and our DHCP renewal process takes 3 seconds. Previous studies reported WLAN ranges of up to 500 meters in radius, and a conventional Linux DHCP usually can take up to 7 seconds.

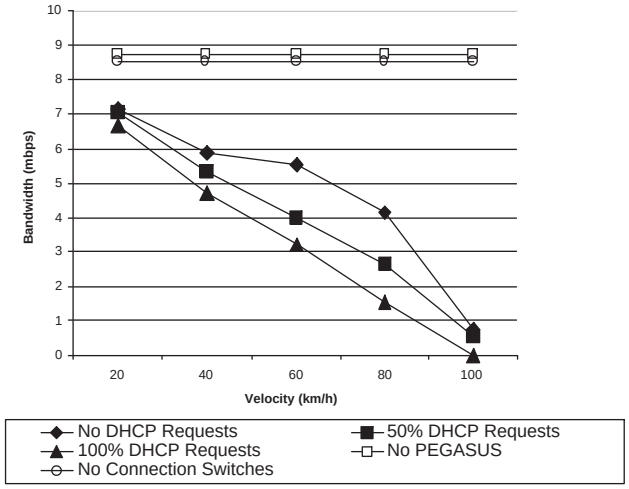

Figure 4.4: Client TCP performance for continuous transfers

The results for continuous TCP transfers illustrated that the connections splitting is not very heavy in overhead. In our lab, clients we able to achieve bandwidth of 8.7 mbs when they did not use PegSvc to proxy their connections, and we measured bandwidth of 8.5 mbs when client connections were routed through PegSvc.

For driving simulations, the chart on figures 4.4 and 4.6 demonstrates the benefits of DHCP cache faster connection transitions. At low velocities the transitions are rare, and the difference in effective bandwidth is not very noticeable, but at higher velocities simulations with 50% DHCP and no DHCP clearly use the access points more efficiently. At 100 km/h tests that required DHCP 100% of the time never completed. The curve for scenarios without DHCP shows a gradual bandwidth decrease with a large dip, for velocities from 80 to 100 km/h. The connectivity period to individual to access point goes from 12 to less than 9 seconds for these cases, and since PEGASUS needs to scan the network when the signal worsens, our scan time starts to be a larger overhead factor. However, the scan can be optimized with a more efficient implementation. Overall, PEGASUS performs very nicely, supporting bandwidth close to 750 kbs even at 100 km/h. At slower velocities, which are more common for urban driving, the bandwidth is a lot higher.

In experiments with shorter TCP transfers, we again show the connection stability that can be achieved with PEGASUS in mobile environment. In Figure 4.5 transfer rates for the shorter segments vary because the transfers are more susceptible to connection transitions. Some of the segments do not experience transitions at all. Nevertheless, PegSvc with no DHCP clearly illustrates the most stable behavior where all of the data is eventually delivered.

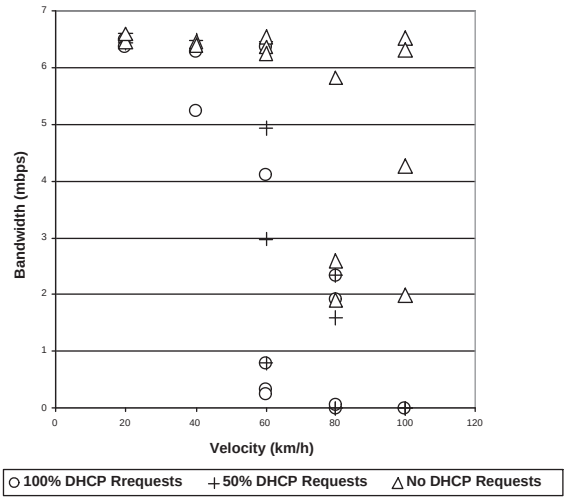

Figure 4.5: Client TCP performance for short transfers

Speed	Bandwidth (Mbps)		
	No DHCP	50%DHCP	100%DHCP
20	7.1377	7.0204	6.6516
40	5.8826	5.3323	4.7181
60	5.5419	4.0092	3.1951
80	4.1477	2.6153	1.5322
100	0.7623	0.5428	0

Figure 4.6: Client TCP performance for continuous transfers - Table

Experiment	Number of Requests	Average Response Time
No PEGASUS	992	79.913
No Switching	808	157.623
No DHCP req (20 km/h)	853	124.34
50% DHCP req (20 km/h)	823	169.788
100% DHCP req (20 km/h)	697	178.34
No DHCP req (40 km/h)	783	141.589
50% DHCP req (40 km/h)	709	173.452
100% DHCP req (40 km/h)	660	192.772
No DHCP req (60 km/h)	851	146.537
50% DHCP req (60 km/h)	487	236.342
100% DHCP req (60 km/h)	435	313.641
No DHCP req (80 km/h)	564	246.221
50% DHCP req (80 km/h)	340	383.855
100% DHCP req (80 km/h)	216	478.855
No DHCP req (100 km/h)	442	267.852
50% DHCP req (100 km/h)	238	336.175
100% DHCP req (100 km/h)	171	410.579

Figure 4.7: Client Web browsing performance

To compare the web browsing simulations we ran them in a continuous loop for a fixed period of time for every connectivity environment. In Figure 4.7 the mobility increases, our results show growing response times, and our client is able to issues less requests in the test time window. However, using PegSvc and DHCP cache obviously provides a significantly slower degradation rate. With PegSvc at 100 km/h we could issue half of the requests compared to the standing still client, and our response time grew by a factor 2 as well. Without PegSvc, the number of

requests went down by a factor of 6, and response time grew 300%.

4.4 Conclusion

PEGASUS is a system that enables wireless connectivity for fast moving vehicles. It provides clients with a constant IP address to preserve application sessions. Furthermore it is able to sustain connectivity in the absense of available APs by using multiple physical interfaces. Efficient connection switching is achieved by storing a global DHCP connection to the PEGASUS server and predicting connection candidates on the client's path. A salient feature of the PEGASUS system is that it does not impose modifications to the infrastructure of deployed networks or protocols. Using in-situ infrastructure and inexpensive dynamic population of the cache helps bootstrapping the service at very low cost. Hence PEGASUS is built to scale with the number of clients.

For WiFi connectivity PEGASUS can use either open access points or WEP/WAP enabled access points with the owner's consent for secured connections. Since all the communication is piped through an encrypted tunnel, PEGASUS offers clients and network operators both authentication and encryption.

We have implemented the system in order to derive its performance parameters. Our experiments showed solid transfer rates and continuous connectivity for high velocity client simulations. The DHCP cache proved to sustain client connec-

tion transitions when the conventional connection renewal schemes degraded beyond workable conditions. We were able to achieve usable and stable network with speeds of up to 100 km/h.

Our current research focuses on PEGASUS buffering module and connection state management mechanisms to improve handling intermittent connectivity. Furtheremore the system's performance under multiple medium interfaces should be further explored.

Chapter 5

Superpeers / scalability

5.1 Introduction

We will soon be surrounded by ubiquitous wireless networks and equipped with devices able to use possibly more than one of them. Protocols and systems should be generic and efficient enough to support a vast array of applications remaining perpetually connected as users move. A major challenge is the proper usage of all available mediums to enhance the provided services in an effective, seamless and non-disruptive way. Scenarios where the users are moving at high speeds, such as when they are inside vehicles, are important, complex, and have attracted great interest in the scientific community. Measurements and ongoing research have shown that WLAN connection for moving vehicles is feasible.

However none of the previous works suggests a solution addressing a complete array of challenges in vehicular WLAN communications. Our system, Pegasus amends this by providing wireless connection roaming at high velocities. To the best of our knowledge, it is the first system that operates over "in situ" Wi-Fi networks,

while at the same time offers transparency to user level applications by allowing a single IP address per user, and does not impose additional requirements to existing infrastructures. Pegasus offers simple deployment, improved scalability, and is the first able to operate over secure "in situ" networks. It remains efficient under intermittent connectivity conditions, and supports heterogeneous network mediums for increased robustness.

In this chapter we furthermore use simulation, based on our implementation to estimate the efficiency and scalability of Pegasus as a function of our decisions for the placement of its components. We use our results to outline possible solutions to the tradeoffs that such system may face. Traditionally, cellular systems use a static, hierarchically layered, structure that divides handovers to micro and macro mobility scenarios. We instead follow an even further decentralized approach that borrows from peer-to-peer experiences breaking down the handover functionality to modules distributed to the system's servers. While all servers bear the bulk of the system's operational cost, a chosen few acting as super peers are in charge of the systems coordination as well as the few lightweight but important tasks. As we explain, our approach is in part because of the already open nature of Pegasus. However, most of our techniques could probably enhance the effectiveness and robustness of systems operated by a single owner.

One of the most important factors affecting the efficiency of any system such

as Pegasus is the effectiveness of access point selection. We formulate this problem using a graph to denote the number of users, the allocated bandwidth from each AP and the cost involved, and study an algorithm to solve this problem using linear programming. This results to enhanced performance. To our knowledge is the first time this analysis is presented.

The rest of the chapter goes as follows. section 5.2 shortly presents the system, 5.3 the AP selection process, 5.4 load balancing, 5.5 a more extensive simulation and 5.6 conclusions and future work.

5.2 System Architecture and Offered services

Pegasus is a system built to deal with rapid 802.11 access point connection switches. It transparently switches between 802.11 access points, or even different mediums (WiFi or cellular) presenting a constant IP and persistent connectivity appearance to users. To support clients we use Pegasus Servers (PegSvc) to manage all connections. Each server may host some or all of the modules required for connection managing, Location Management, as well as authentication, authorization and auditing functions. However, both the prototype and the simulation used in this chapter refer only to the WiFi part of Pegasus and the core server functions of forwarding traffic, selecting APs and alloting connections to the mobile users .

To efficiently manage all system functions, we rely on the set of PegSvc orga-

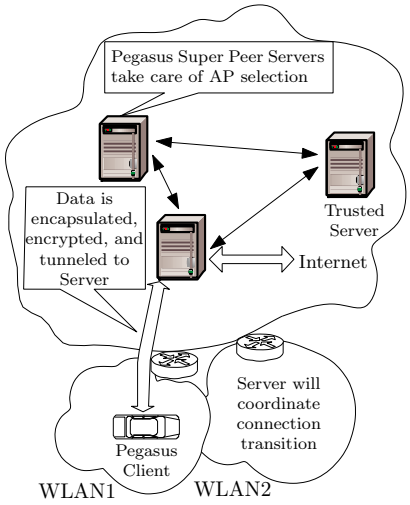

Figure 5.1: Pegasus system architecture

nized in a peer to peer network. A small set of these machines are exclusively used for the system and are trusted. All others belong to the users and share most of the system load, mainly tunneling user traffic. A few user machines are selected to act as super peers. These are computers with high up-time and enough resources to take on the task of being responsible for keeping track of the user status in a given area each. The scope of this chapter does not extend to policies that would lead users to contributing to the system in order to enjoy its benefits. However auditing and authorization is straitforward, and could employ any of the policies described in recent papers as described in the related work area.

Our system's efficiency is based on re-using connection knowledge among many clients and pre-fetching connection candidates on the client path to minimize connection setup overheads. The overall Pegasus architecture is designed to support

very large networks with WAP security to protect both system clients and network operators. In addition, PegSvcs hosts a connection database and manage proposed APs to aid clients as they move.

In this chapter, we assume that our clients will travel in a more or less connected grid of WLAN connection spots, and they will be able to find an available WLAN network most of the time. Since PEGASUS operates on either open and secure APs with the consent of the access point owner, the assumption is realistic. The non-connectivity periods should be relatively brief, and PEGASUS can to fall back to non WiFi wireless network if need arises.

We achieve efficiency by reusing a DHCP cache globally. Pegasus caches all of the DHCP connections from all of the clients in a global cache, and continuously reuses them. Since DHCP is bound to the client MAC address, in order to reuse DHCP connections, users change their MAC address to a value handed-in by the PegSvc. Once a user moves on to the next connection, it changes the MAC address again. This concept of recycling acquired DHCP connections and using a different DHCP identity at each independent access point island is the core concept that allows Pegasus to achieve its efficiency and scalability.

Every WLAN is independently managed, possibly connected through multiple ISPs, private address spaces and NATs. As depicted on Figure 5.1, to handle such heterogeneity, client's traffic is tunneled to a PegSvc. This server can be operated by

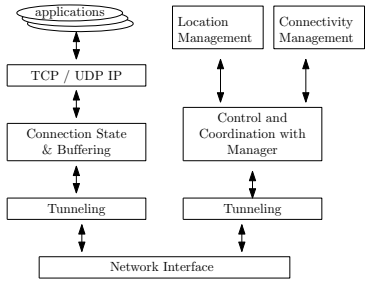

Figure 5.2: System Architecture: Mobile Client

a third party and acts as a multiplex point for all client Internet communications. PegSvc attempts to predict the client movement through deployed WLANs and offers choices for the next access point connection. The switch from one AP to another will not sever the ongoing client application sessions. Moreover, since the server acts as fixed peer to the non-mobile connection endpoints, it buffers network packets to smooth possible connectivity dead spots. All of the tunneled traffic is encrypted to offer extra security for the client data.

Pegasus offers transparent connectivity appearance to client applications, deployment on top of "in-situ" access points, without managing or changing the existing infrastructure, support for any WLAN configurations, and simple installation on clients, without modifications to the existing operating systems and applications. It supports existing user equipment without requiring any custom or specialized hardware at the client or server devices and provides utilization of multiple network

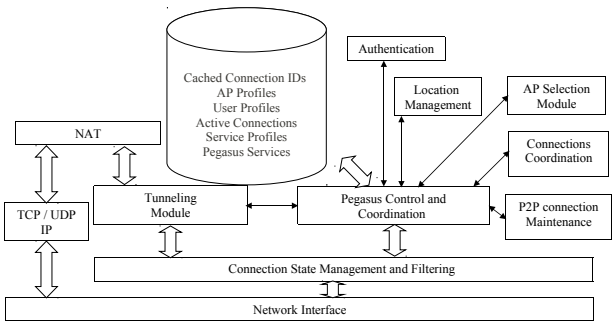

Figure 5.3: System Architecture: Pegasus Server Side

mediums available at the client, extensibility for future performance enhancements to further improve mobile connectivity. It offers simple system deployment and dynamic growth of the system connection database, and can scale to support a growing number of clients.

One major advantage of Pegasus system is that the selection of future Access Points is performed in the super peers. In this way, the system load can be better balanced and the bandwidth provided can be maximized without overloading the APs. Furthermore, APs proposed should be in range of the user for larger amounts of time, thus avoiding unnecessary handover costs.

In this section we address the problem of efficiently selecting APs. Given the available APs and a number of users in the system, the question becomes how to direct the users to APs so that each user's bandwidth requirements are met, the handovers are minimized, and the APs are not overloaded. We formulate this problem to be solved using quadratic programming and rely to each super peer responsible for handling an area of the system to provide a solution fit for the traffic it experiences.

In Figure 5.6 we see how a toy example of a single road segment covered by 3 APs might look like. Notice that the segments between points $(1, 2)$, $(3, 4)$, and $(5, 6)$, users have no choice but to connect to AP1, AP2, and AP3 respectively. However in the segments $(2, 3)$ and $(4, 5)$ they have more than one options. In fact, while passing through the latest segments, they should switch APs.

name	denotes
e_{ij}	Edge between i and j representing (i, j) segment.
APx_{ij}	Edge between i and j due to Access Point x.
$c_{x_{ij}}$	Bandwidth of APx assigned to APx_{ij}.
$c_{x_{\text{total}}}$	Total bandwidth provided by APx
u_{ij}	Number of users traveling from i to j.
u'_{ij}	Estimated number of users traveling from i to j in the next time period.
$c_{\text{usr}_{ij}}$	Assigned bandwidth per user in e_{ij}
$c_{\text{total}_{ij}}$	Maximum possible bandwidth from all APs in e_{ij}

Figure 5.4: Access Point selection: variables

5.2.1 Bandwidth Allocation Graph

Based on this observation we consider the graph of all nodes 1 through 6 as shown in Figure 5.5 in which we allow multiple edges between nodes. We consider the maximum continuous segments in which AP coverage is constant, then create one node for each end of those road segments. For each APx that covers the path segment from node i to node j, we create an edge denoted in as APx_{ij} and are allowed to allot a part of the APx's bandwidth. We use the variable $c_{x_{ij}}$ to denote the assigned bandwidth. Then the total bandwidth in any AP cannot exceed some what this AP can support. In fact, in order not to overload any AP, we require that the total bandwidth over a time period is only part of the available. In general, we constrain $\sum_{ij} c_{x_{ij}} < c_{x_{\text{total}}}$.

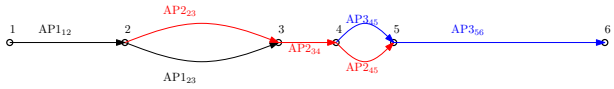

Figure 5.5: Access Point selection: single road paradigm - Part of the derived graph

4

5.3 Access Point selection

5.3.1 User Graphs and the cost of handovers

To predict the bandwidth available to any user in a segment, we also need to know the number of users in this road segment. Therefore we consider a second graph having the same edges, but in which nodes i and j are connected by the edge u_{ij} iff there is a single segment from i to j. We set the weight of u_{ij} to be the number of users in that segment and call this graph the User Graph. Similarly we consider v_{ij} to be the average speed of those users. We can now estimate the bandwidth available to each user as $\frac{\sum_x c_{x_{ij}}}{u_{ij}}$. Given the position of all APs and users, we could force users to switch to those APs that optimize for this quantity, providing effectiveness, or efficiency, or a balance between those two.

Pegasus periodically re-computes the values for $c_{x_{ij}}$ and uses it to assign users

to APs. Therefore, instead of using u_{ij} and v_{ij} , we will use u'_{ij} and v'_{ij} which is an estimation about the number and speed of users in e_{ij} traveling from i to j during this time period. Since estimating the future system state is not this chapter's main goal we will here use a greedy, simple to implement, strategy with minimal computational cost. More complex and accurate predictions would results in better estimating the u'_{ij}'s, and the handover cost, however the method of the analysis presented would not change. We use a graph to keep track of a time weighted average of users for each segment. We call this Estimated Future User Graph and it consists of the User Graph but with the edges having weight that is computed every cycle as :

$u'_{ij_{\text{new}}} = \lambda \cdot u'_{ij_{\text{old}}} + (1 - \lambda) \cdot u_{ij}$, and

$v'_{ij_{\text{new}}} = \lambda \cdot v'_{ij_{\text{old}}} + (1 - \lambda) \cdot v_{ij}$, for $\lambda = 0.9$

The next step in our analysis is to take into account the cost of handovers in the system. We need to address when and where each of the users should switch AP, and what this new AP should be. However, it is impossible to know the exact cost of future handovers since these will be affected by the path and speed of each user. The best we can do is try to predict this cost based on our estimations about the future state of the system. Let us first consider a simple case such as the one of node 2 in Figures 5.6, 5.7. In this case we already know that all users

passing through $e_{1,2}$ will also pass through $e_{2,3}$, since this is an example where a segment covered by a single AP is followed by a single segment covered by that AP as well as additional APs. Initially, all users were connected to AP1, however part of them will switch to AP2. Even with Pegasus optimizations in place, where most of the time a pre-existing connection is suggested, their connection quality will suffer. The total cost in the segment will depend on how much traffic from AP1 needs to be taken over by AP2, but also the speed of the users and the length of the segment. In the following paragraphs, we see that we also have the constraint $c1_{1,2} > c1_{2,3}$. Thus we can have the cost in the case of in Figure 5.5 segment 1 to 2 to be: $f(u_{1,2}, v_{1,2})(c1_{1,2} - c1_{2,3})$.In the general case where e_{ij} is followed by only e_{jk} we could use the equivalent $\frac{1}{2}f(u_{ij}, v_{ij})(\sum_x |c1_{i,j} - c1_{j,k}|)$. In order to use convex programming however, we will use

$$\frac{1}{2}f(u_{ij}, v_{ij})(\sum_x (c1_{ij} - c1_{jk})^2)$$

In Figure 5.7, node 3 reveals a case where handovers from users coming out of e_{23} towards 4 might be connected to either AP1 or AP2. The cost using this formula will be estimated to $\frac{1}{2}f(u_{2,3}, v_{2,3})((c1_{2,3} - c1_{3,4})^2 + (c2_{2,3} - c2_{3,4})^2 + {c3_{3,4}}^2)$

When considering the general case, we need to take into account that e_{ij} might lead to more than one segments. As an example, consider Figure 5.9 node 3. Users

coming from $e_{2,3}$ may continue to any of $e_{3,9}$, $e_{3,15}$, or $e_{3,10}$. Therefore, in order to know exactly the handover cost we would need to know which path the users will follow in the future. To get an estimation of this cost we use the Estimated Future User Graph. In particular we estimate that from the current u_{ij} users in e_{ij}, $u_{ij} \cdot \frac{u'_{jk}}{\sum_k u'_{jk}}$ will move to e_{jk}. Therefore the cost of handovers in each segment will be:

$$\text{cost}_{ij} = \frac{1}{2} \cdot \frac{u'_{jk}}{\sum_k u'_{jk}} \cdot \sum_x (c1_{ij} - c1_{jk})^2$$

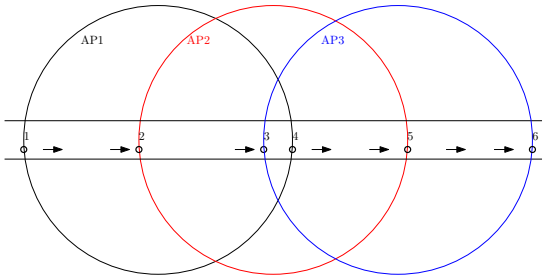

Figure 5.6: Access Point selection: single road paradigm

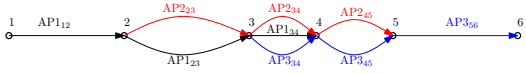

Figure 5.7: Access Point selection: single road paradigm - Part of the derived graph

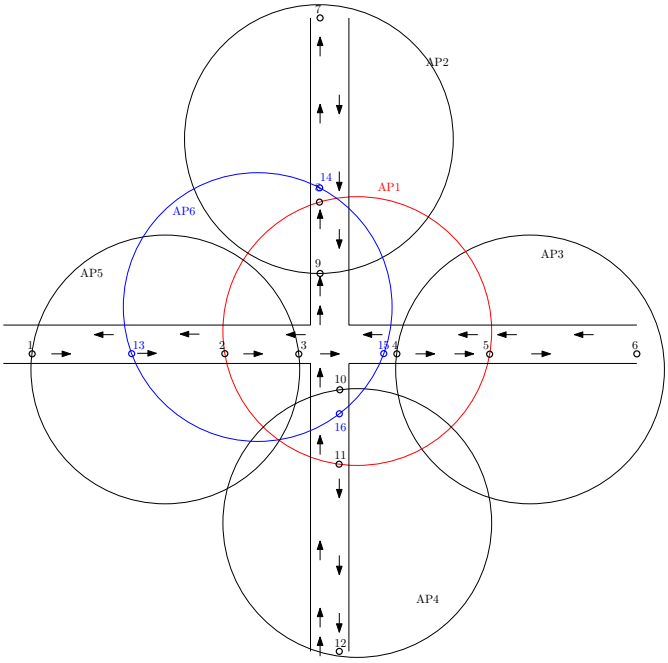

Figure 5.8: Access Point selection: intersection of two roads

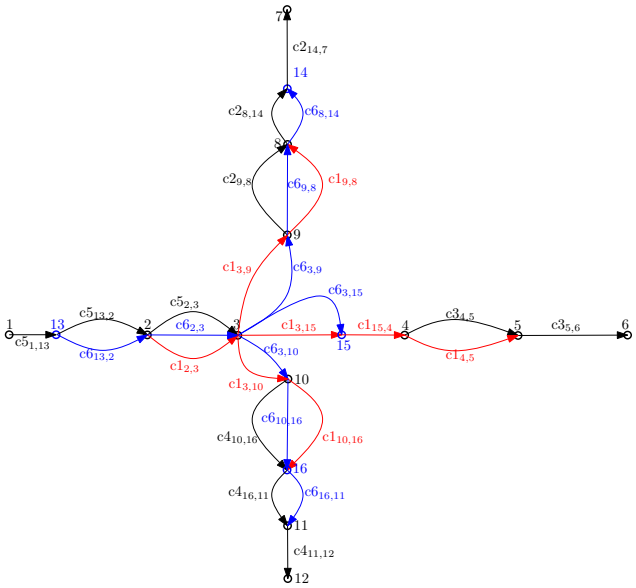

Figure 5.9: Access Point selection: intersection of two roads- Part of the derived graph

5.3.2 Optimization and additional constraints

We could at this point set our optimization goal. The problem is that we want to both maximize the available bandwidth for every user, and minimize the number of users that need to handover. Additionally, although ideally we would want bandwidth to be shared among all users, that might be impossible because some road segments will always have less users and/or more available bandwidth than others. It would therefore be inefficient to refer to the minimum available bandwidth per user, since this would not have any effect on the optimization of richer areas.

In essence we care about the quantity of bandwidth per user,

$$c_{\mathrm{usr}_{ij}} = \frac{\sum_x c_{x_{ij}}}{u_{ij}}$$

which is upper bounded by the max available bandwidth of all APs present in the segment over the number of users in that segment

$$c_{\mathrm{total}_{ij}} = \frac{\sum_{x:\exists \mathrm{AP}_{x_{ij}}} c_{x_{\mathrm{total}}}}{u_{ij}}$$

.

Fairness then requires to add a constraint for every segment e_{ij} where the max

available bandwidth per user:

$$\forall i, j, l, m : c_{\text{total}_{ij}} \geq c_{\text{total}_{ij}} \Rightarrow c_{\text{usr}_{ij}} \geq c_{\text{usr}_{ij}}$$

Having added these constraints, to maximize the bandwidth for all users we will press the optimization function to reward allocating available bandwidth to areas where the available bandwidth per user is less. Thus, instead of aiming to maximize the total c_{ij} for all i, j, we will weight by the max available per user bandwidth and maximize the quantity:

$$\sum_{ij} \left(\frac{u_{ij}}{c_{\text{total}_{ij}}} c_{ij} \right)$$

Taking into account the cost of handovers, we can therefore try to maximize the quantity

$$w \cdot \sum_{ij} \left(\frac{u_{ij}}{c_{\text{total}_{ij}}} c_{ij} \right) - (1 - w) \sum_{ij} \text{cost}_{ij}$$

Where w needs to be adjusted based on our experimental results about the cost of handovers in the system.

5.4 Load balancing amongst super peers

Any Pegasus server responsible for Access point Selection over an area, needs to perform a number of demanding tasks such as track the locations of all users in that area, decide when and where to handover, and suggest new Access Points. We therefore balance the load by using multiple servers we call super peers, each of which is responsible for part of the area. Most Pegasus servers are used for tunneling, while tasks like AP selection are reserved for super peers, A few, trusted servers organized in a chord ring are used for authentication, authorization, and auditing. Furthermore, a subset of the rest of servers with is selected to be super peers.

Super peers are organized in a hierarchical, distributed k-ary tree as in . The ones on the root group are also known to one of the trusted servers. Each group has a leader that in case of failure of any other server will assign a new server and redirect queries and updates to it. In case of leader failure, the node that points to the group will take care of recovery. Each super peer also keeps pointers to others with which it shares nodes. Therefore, if e_{ij} is managed by a server, any e_{jk} will either be managed by that same server, or that server will keep a pointer to the server that manages it. As a result, users on that path will be able to know what the next super peer managing the road server should be. belongs t Each super peer is responsible for bandwidth allocation in an area of the system. It needs to keep

track of the number of users in that area, suggest new APs to them and periodically run the AP selection algorithm in that area. When the number of users in an area and therefore the load on a server is increased beyond its capacity, it needs to split the area it is responsible for. To that end, it computes the user mass center considering each segment e_ij as a point (x_ij, y_ij) located at its center. Thus, let $(x_{\text{ctr}}, y_{\text{ctr}}) \Leftarrow (\frac{\sum(x_{ij} \cdot u'_i j)}{u'_{\text{total}}}, \frac{\sum(y_{ij} \cdot u'_i j)}{u'_{\text{total}}})$ and then let us consider all the splits along the line defined from $(x_{\text{ctr}}, y_{\text{ctr}})$ and the center of any other edge (x_k, x_l). These will split the segments into two sets ($E_{split_k,l}$ and $E - E_{split_k,l}$) of roughly equal number of users (roughly because the users of approximately one edje will not bew split)

$E_{split_k,l} =$

$\{e_{ij} : (x_{\text{ctr}} - x_{k,l})(y - y_{k,l}) - (y_{\text{ctr}} - y_{k,l})(x - x_{k,l}) > 0\}$. For each split let us consider the total bandwidth of all APs that have segments in both subsets and call it cut_(k, l). Then the super peer will choose to split edges into two subsets so that cut_bw(k, l) is minimum in total linear to the number of edges time, and then assign one of these subsets to another super peer. If the two subsets were connected, the servers keep a connection, and so they can directly point any users moving between segments to the right super peer.

5.5 Simulation

5.5.1 simulation setup

We simulate the movement of vehicles on a $2km^2$ area of Washington DC city center with cars distributed through it. For each scenario we have 40 different configurations of 10 simulation minutes, with 200 vehicles and a transmission range of 120m. For the city environment the nodes minimum speed is 18Km/h and the maximum is the maximum allowed on that specific road based on the data provided by the Topologically Integrated Geographic Encoding and Referencing system of U.S. Census Bureau. The scenarios follow a realistic mobility pattern generated with the VanetMobiSim [HFBF06] tool. Each generated scenario has a number of APs placed randomly. All experiments keep the same basic configuration but the number of and locations APs is random. Per average we allocate 40 APs but the number varies up to 100. Our main interest is to see how a future system would behave under realistic conditions and if there are any possible gains from our optimized approach.

We use the Computational Geometry Algorithms Library (CGAL) to pre-process the NS2 files produced by VanetMobiSim as well as to generate random numbers. For each user, we keep track of all the Access Points that he will ever meet and create an event file containing this information. Based on the files from all

the users, and the strategy that the user will employ to decide which APs to connect to, but using information available at the time to the user, we create another event file that lists all choices about handoffs. We also keep a file listing the number of users in every AP at any time. We use these files to quickly simulate what the resulting bandwidth of each particular will be. Our prototype implementation serves to validate the observed bandwidth we predict.

5.5.2 simulation results

APs

In Figures 5.10 and 5.11 we see the average distribution of the number of users per AP during simulations. We exclude APs that have zero users from these figures, simply because our interest is related to the load Pegasus imposes on the APs.

We also show the 5% and 95% cutoffs for all observed values. Due to the different positions of each AP, each one has to deal with a varying amount of load. As we see, using the optimized approach reduces the load to many of the APs that are contributing to the system. This allows for fair resource utilization.

Bandwidth

Figures 5.12 and 5.13 depict the bandwidth distribution of the users as a percentage of their total time in the system. Since users are equal, any two traveling

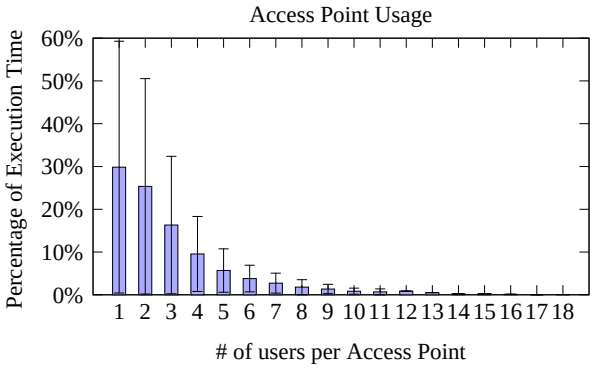

Figure 5.10: Access Point usage, greedy

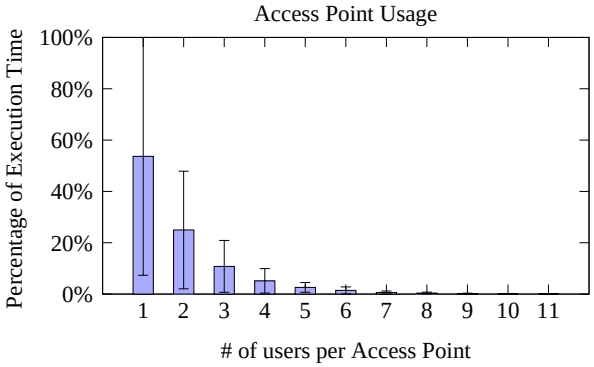

Figure 5.11: Access Point usage, optimized

for enough time would result in the same distribution and any cut of intervals will depend on the simulation time. Therefore we do not show any intervals in this picture which gives us just the average bandwidth distribution.

These two figures clearly show the gain our system can provide to its users. This is because for the same user velocity, it provides faster handoffs with DHCP caching , but also because it allows for an informed AP selection.

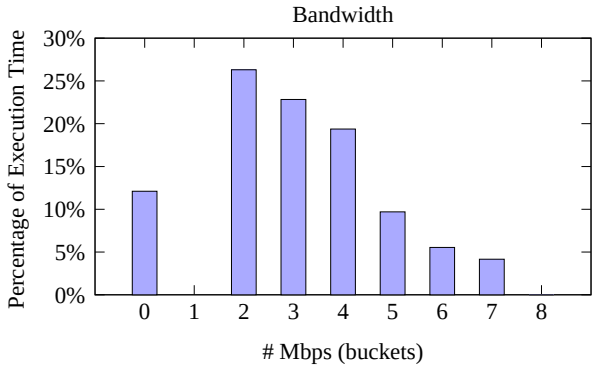

Figure 5.12: Average Bandwidth available without using the system

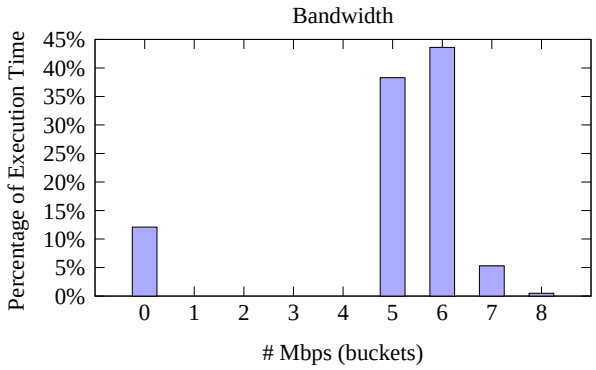

Figure 5.13: The system's Average Bandwidth as a function of time for any user

5.6 Conclusion

In this chapter we present Pegasus, a system providing wireless connection roaming at high velocities. While WLAN over "in situ" WiFi networks connection for moving vehicles is feasible, a solution should address a complete array of the challenges in vehicular WLAN communications. Pegasus provides a solution offering simple deployment, improved scalability, and is even capable to operate over secure "in situ" networks. It is based on user participation, and can be tuned to offer

participating users benefits according to the resources they provide.

We propose how user connectivity can be enhanced on our system and present an implementation of Pegasus for one user and a larger simulation for 200 users on the D.C. Washington area using realistic mobility patterns. We show that per average the system can sustain rates of over 5 MBps for over 87% of the time. Thus Pegasus, can provide efficient mobility support on environments of small connectivity range.

Chapter 6

Virtual Access Points for Wireless Communications

6.1 Introduction

Consider a world where users are equipped with mobile devices of varying capabilities, with varying processing, sensing and communication capabilities. These will be able to enable a vast number of applications. Users could access the latest news and information, communicate, and even publish information, or locate and access wanted resources. Information about interesting phenomena, traffic congestion, or health status could be monitored from and disseminated to users in vehicles to allow us to gain understanding, save money and lives.

In the future, pervasive wireless world, all roads and cities may be covered by roadside base stations and access be provided to both pedestrians and vehicular users. However, the cost and the capabilities of the available wireless mediums will vary greatly. It is reasonable to assume that there will be zones covered by

interfaces with high bandwidth, and low cost, yet limited range of connectivity, as well as pervasive means to connect at a lower bandwidth and higher cost. For the moment, road side equipment or Access Points (APs) are not always present, resulting to uncovered areas where the only possible communication mode is from one vehicle to another. Furthermore, equipping each new generation of networking access devices requires time to be deployed in large scale. User equipments are easier to update and will often have more capabilities than the roadside infrastructure.

As the value of the offered services increases, so does the cost of a possible failure of the infrastructure that provides it. Maintaining communication capabilities in disaster scenarios is a crucial factor for avoiding preventable loss of life and damage to property [TM05], and in a world of ubiquitous connectivity, it will be increasingly important. During a catastrophe such as an earthquake, power outage or flooding, the main wireless network structure can be severely affected. Reports form September 11th point out communications failures contributed directly to the loss of at least 300 firefighters and prevented a good management of the rescue efforts what could have contributed to the loss of many other lives [oTAUtUS04, MC02]. Moreover, communication failures are pointed as one of the obstacles in the co-ordination of the rescue resources in the 1995 Kobe earthquake [LUKY96]. These failures further prevented outsiders from receiving timely information about the severity of the damages. These communication breakdowns delayed the relief efforts what could

have prevented the loss of numerous human lives.

Furthermore, "historically, major disasters are the most intense generators of telecommunications traffic" [TM05]. The public communication networks, even when available, may fail not only because of physical damage, but also as result of traffic overload. Therefore, the regular public networks alone are often not sufficient to allow rescue and relief operations [TM05].

When considering devices with multiple interfaces, some of which are out of range of any base station, as well as catastrophe scenarios, it makes sense to distribute the responsibility of disseminating information among the mobile nodes. This chapter presents the Virtual Access concept as an effective method to increase the network coverage. More precisely, the main focus of our work is twofold: our aim is provide a technique to use available short range wireless interfaces to extend coverage to nodes outside of covered areas and even more importantly to allow for the dissemination of stream traffic that could be used, in disaster scenarios, to coordinate rescue teams and deliver general information to nodes. We consider scenarios with mobile nodes in a city environment receiving stream traffic through fixed Access Points (APs). This kind of architecture, suited for daily usage, is extremely sensible to disaster scenarios. We discuss Virtual Access Points(VAPs)[FCF+08, CFF+08, CFF+09], a simple, yet powerful, technique to extend coverage to nodes outside covered areas. We examine dissemination of data using VAPs and then fo-

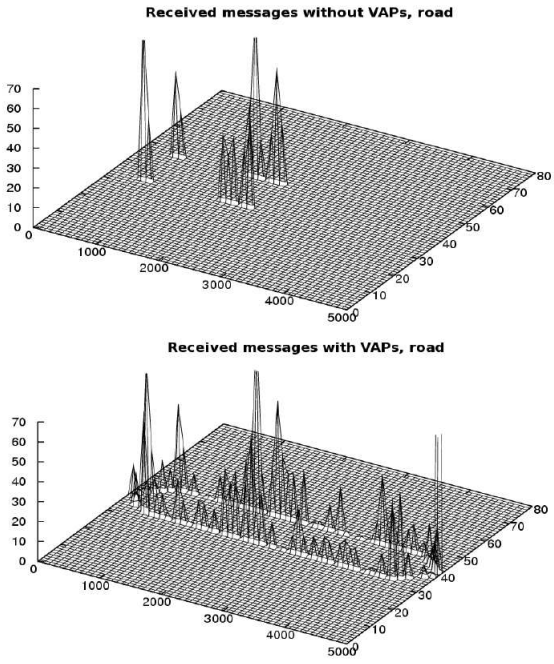

Figure 6.1: Typical receiving messages map for a 5 APs road scenario

cus on streaming data dissemination in a city environment. We discuss how, and to what extent, VAPs can enable streaming on a highway 6.1 or a city 6.2 environment. Then we consider three distinct catastrophe scenarios, namely earthquake, flooding and power outage, and the case of APs random failures.

The rest of the chapter is organized as follows; Section 2 discusses related work, whereas Section 3 formulates the proposed solution and outlines our analysis. Section 4 presents the VAP proposal applied to a series of disaster scenarios as a way to keep the information flowing in a network even during a catastrophe. Section 5 presents the simulations results, and, finally, concluding remarks and future research are given in Section 6.

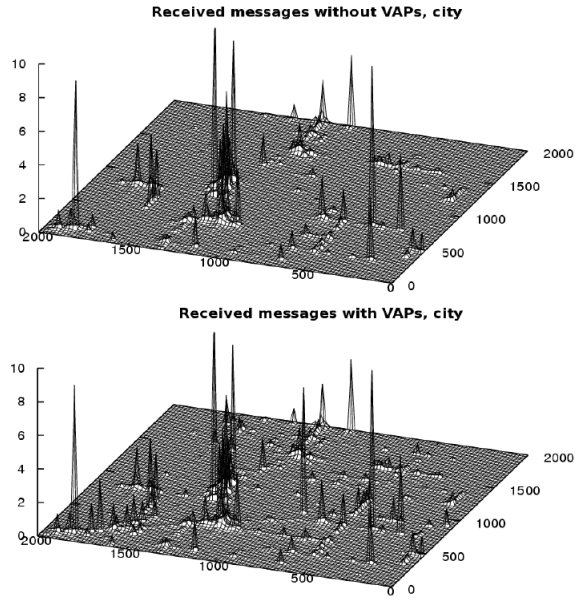

Figure 6.2: Typical receiving messages map for a 50 APs city scenario, we can see how VAPs allow us to connect existing "connectivity isles"

6.2 Related work

Vehicular communications constitutes an especially active area. Various techniques and target applications have been studied. Most of them target the problem of improving safety through the use of wireless networks. Therefore, many proposed techniques prioritize critical messages while allowing non critical applications to utilize overhead bandwidth. The system considered uses overhead bandwidth for delivering streaming data, often requested by more than one user. The network architecture followed here is based in the basic multiple Infostation model [GBMY97], However, we will refer to the Infostations as Access Points (APs).

An appealing solution a similar problem, but more expensive and probably

harder to implement, is the system proposed by Gavrilovich in [Gav01]. In this work, to compensate for the velocity of the cars, the authors propose the creation of a chain of mobile APs on the center of the highway. Their role is to increase coverage and compensate for the high velocity of cars on the highway. This system may not be a solution to any of the disaster scenarios considered.

Data dissemination is a key issue in any network, and has been well studied. It is push-based when the data are repeatedly broadcasted in a cycle; pull based when it is explicitly requested by the users; and hybrid when using a mixed push-pull approach. Push or push-pull based delivery is attractive in many systems and has been well studied [HFBF06, AFZ97, KM04b]. Environments of mobile users with deadlines are discussed in papers [FVDN04, TN06].

Communications infrastructure plays a critical role in all phases of disaster prevention and recovery [TM05]. However, due to both the nature of disasters, and the fact that communication networks are often not designed to operate under arbitrary load conditions, it is possible they may fail when needed the most [oTAUtUS04]. In such situations, often the number and density of users that need to use the network is unusually high. Furthermore, part of the infrastructure can be unusable.

A number of solutions has been proposed that makes use of ad hoc protocols [KR06, DMJR05]. These rely on an ad hoc network or use a hybrid scheme. In the case of hybrid systems, we see the ad hoc used to extend the reach and/or

increase the capacity of the fixed network infrastructure. However, no provision is made for cases where the network is fragmented. Furthermore, the transparency of the fixed system is violated and only nodes capable and configured to connect to the ad hoc part of the network can take advantage of its existence.

This work aims to extend access to areas out of AP coverage. In the same line, the SPAWN, introduced by Gerla et al. in [DNP+04, NDP+05], where it is discussed how vehicles should interact to accommodate swarming protocols, such as BitTorrent traffic. In SPAWN, the nodes passing through APs collect data that they subsequently exchange among nearby nodes. SPAWN focuses on a restricted application that generates great volumes of traffic. Nodes are often required to carry traffic useless to them and the BitTorrent protocol is bandwidth intensive. Moreover, the number of retransmissions of a message in a vehicular network is estimated to be approximately 3 and so our gain from using the swarming protocol in this environment is non-optimal.

Disruption and possibly delay tolerant networks can overcome the problem of sporadic lack of direct communication, however, most are not tuned to work over disaster environments. The Data Mule project [SRJB03] and the Message Ferrying scheme [BTAZ06], designed for sensor networks, propose the use of mobile nodes to collect data from the sensors, buffer it, and deliver the collected data to a sink. As opposed to these works, we consider the problem not of retrieving data from

the nodes, but of disseminating it to them. Our goal is not to build a full delay tolerant system, but merely to extend the capabilities of existing APs with the use of VAPs [FCF+08] in a way that will allow the system to survive a catastrophe. The MULEs (Mobile Ubiquitous LAN Extensions) and ferries utilize nodes navigating through the sensor network to collect data in 'mobile caches'. According to the Data Mule project, all the nodes are fixed and only the cache is mobile. In contrast, in our scenario all nodes are mobile but we cannot affect their trajectories. Message Ferrying also considers mobile nodes but in that approach, as well as in [LR00] and [BH00], the nodes are required to follow specific paths and even move in order to help message delivering. The work presented in [BH00] proposes a multicast protocol for the highway environment where information dissemination though message flooding for VANET environments is proposed. Our proposal advocates using of a more systematic bandwidth efficient approach for data dissemination.

MaxProp [BGJL06] is a disruption-tolerant network base on prioritizing both the schedule of packets transmitted o other peers and the schedule of packets to be dropped. It makes use of several complementary mechanisms, including acknowledgments, a head-start for new packets, and lists of previous intermediaries. It is not transparent and is tuned to serve the needs of a vehicular delay tolerant network for connections between its users.

Chen et al. [CKV01] study network delay as a function of the number of cars

and their velocity. The authors note that node mobility on highways can improve end-to-end transmission delay when messages were relayed. Furthermore, that low density networks may experience higher delays. These results are directly related to our work. VAPs locations should be selected so that information is not too widely spread and messages of time sensitive applications can reach their destinations in time.

The network architecture we followed is based in the basic multiple Infostation model [GBMY97]. However, we will refer to the Infostations as Access Points (APs). Vehicular communications constitutes an especially active area. Various techniques and target applications have been studied. Most of them target the problem of improving safety through the use of wireless networks. However, in most cases, the vehicular networks considered are not tuned to help in general disasters like an earthquake or flooding, where part of the roadside equipment may be damaged and furthermore, the needs for communication are different than that of accident prevention. In our scenarios the traffic can be originating from a source possibly located outside the network that needs to disseminate information. We could use overhead bandwidth of a VANET for delivering streaming data, often requested by more than one user.

6.3 Virtual Access Points for mobile nodes

6.3.1 Protocol

The main focus of the Virtual Access Point technique is to decrease the areas not covered by roadside APs so as to minimize the problem of intermittent access to mobile nodes. If we are able to decrease this problem, then stream traffic for mobile users may be enabled.

Each node, after receiving a message, caches it and can in future become a VAP, acting similarly to a relay node. Note however that instead of just resend the messages the VAP, stores the message and may send it more than once or not at all depending on the caching strategy and depending on the locations it will pass by. VAPs strive to supplement the lack of real APs in a given area broadcasting messages received previously from other AP or even VAPs. A node acts as a VAP if it is not in range of neither an AP nor a VAP and its distance from the nearest AP is $2r$, where r denotes the AP transmission range. This in practice means that nodes are allowed to act as VAPs only when it is in a distance where its MAC layer does not detect any APs above a very low SNR. We also assume that the MAC layer resolves conflicts to the medium access.

In case a node senses another node acting as VAP in the same region, it gives up being a VAP, even if it lies in the area it could act as one. Therefore, the first

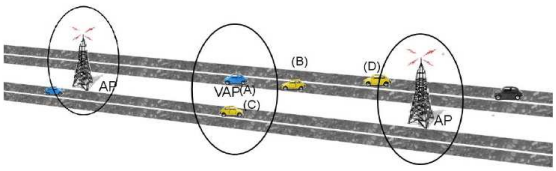

Figure 6.3: A road coverage vision

node to broadcast VAP messages in a given region becomes the VAP . Nodes are not allowed to act as VAPs during two consecutive time intervals. Fig. 6.3 shows a typical scenario where a vehicle (A) acts as VAP providing access to vehicle (C). In turn, vehicle (D) will transmit the information it just received from the AP when it reaches the same area.

6.3.2 Analysis

At each second 1, 2 or 3 packets are generated from a source and spread through all antennas, which are then in charge of broadcasting the stream message to their neighbors. Each message is transmitted from each antenna just once. Every mobile node has a limited size buffer where it stores the last received messages. During cache replacement the oldest stream message, with lower stream ID, is discarded first, regardless it was the last one to be received or not.

The system is a best effort one; there are no guarantees that every node will receive all stream packets, but using VAPs, we aim to increase the chances for timely reception. Note that the VAPs increase significantly the overall number of messages

in the network; however, this increase occurs in areas with no previous coverage, so they create no significant interference with the normal network behavior. The increase in the number of messages sent is upper bounded by:

$$\#IncMesg \leq nEM = nVAP * BS \quad (1)$$

Where $\#IncMesg$ is the number of increased exchanged messages, nEM is the maximum number of exchanged messages, $nVAP$ is the number of Virtual Access points, and BS is the cache Buffer Size. Unfortunately not all received messages are useful for every node and the duplicate or old objects are discarded. The number and locations of the VAPs will greatly affect the system's performance. consequently, the role of VAP is assigned dynamically. Based on the node's mobility pattern and distance from any APs, the nodes autonomously decide if they should act as a VAP.

6.3.3 Formal Verification

The used technique

To analyze and evaluate our proposal we used the methodology proposed in [CLF07]. The method uses Model Checking to determine if a given model M, presents a defined property P. Both, M and P are provided by the protocol designer and precisely defined. M is composed by the finite set of variables V, $V = v1, ..., vn$, the set of initializations I, where it is applied $I(V)$ or I is a condition over V, and a set of transitions T, $T(V, V')$, where V' is the new value for the variable V after the

application of the model step. The model checking tool uses then M, to build the set of all possible system states. We selected the SPIN model checker, which creates a binary decision diagram to represent to represent the model. Let $G = (V, I, T)$ be the set of all states, and $P = (V)$ the property to verify, the tool must than search if P can be satisfied starting with I and applying T a finite number of times. If M model all the possible relations, than G contains all the possible outcome system states.

The technique is based on ground principles, such as the abstraction of network topology. When creating a model of the protocol, the designer should abstract the topology and focus on the possible network relations. Any proof that takes into account a specific topology, will only prove the behavior of the protocol for that particular evaluated topology. For the set of properties we want to verify, the topology is not a relevant factor. So, instead of enumerating all the infinite possible topologies, it is better to avoid it and focus on the node relations. In order to do that, the method proposes the use of three kinds of nodes, namely, source (S), destination (D) and intermediate (N) nodes. Let R, $R = (Origen, Destination)$, be the set of all possible relations among the different kinds of nodes, $R = \{(S, D), (S, N), (S, Null), (N, S), (N, N), (N, D),$
$(N, Null), (D, S), (D, N), (D, Null), (D, N)\}$, where $Null$ represents the case where the message is lost or corrupted. If the model M contains all the relations defined

in R, and R contains all the possible and relevant message forwarding cases, then all the possibilities will be presented in G, and thus in consequence in the verification. With this process the topology becomes irrelevant since the relations among intermediate nodes will model the possible relations, not the network topologies. Consider $C \subset V$, where $C = \{c : c \in M \textit{ and } c \textit{ controls the message flow}\}$. C is the set of all procedures used by the target protocol to control the flow of information through the network, for example, c could be the last received message counter. Assume that $\forall c \in C, c = \{0, 1\}$, if $I(C)$ is random, the tool will generate all possible initial values, each of the two possible values for each c will be represented in G. If G contains all the states, the search for P over G is granted even without considering the topology.

Another principle is that the service of the layers, apart the one where the algorithm is in, should be abstracted, since they are not the object of study. The services provided by the lower layers should be modeled as available and trustable, unless some cross layer aspect is crucial for the protocol validation. This step consists of the application of the assume-guarantee paradigm [GL94, Pnu85], since assume that the lower layers behave as expected and, if it is required we need to guarantee this assumption through verification.

One of the key aspects to enable the proper verification is the modeling of the communicating channel. The channel should be available in a random way among

the three defined kind of nodes or no node at all. Let K be the set of packets defined in the target protocol, $K \subset V$ and $K = \{\forall k \in K : k \rightarrow O \vee k \rightarrow N \vee k \rightarrow D \vee k \rightarrow Null\}$. Where the relation $\rightarrow$ stands for "may be delivered to". This means that any node, or even no one at all, may receive the packet. This guarantees that all the relations will be verified in the end.

The protocol behavior must also be divided into internal and external behavior. This is an instance of compositional reasoning [BCC98]. Composition is a divide and conquer approach [STS+07] where the target system is divided into small components that are verified separately. However, it is important to notice that the composition in this case refers to the division of the protocol behavior and not functionalities. The protocol behavior should be divided into internal and external, and each part verified separately. Internal behavior refers to how the protocol handles data and controls messages internally to the node. The external behavior refers to how the whole network reacts to the messages. Both models should be independent and modeled in such way that the internal behavior could act as a procedure of the external behavior.

Every information regarding the verified protocol should be modeled as a variable and, as far as possible, randomly initialized (e.g., have already received this packet or is the node inside or not of a VAP zone). When possible, such information should also be modeled with boolean variables. This is a form of abstrac-

tion [CBKK94], since we will not be representing the whole set of possible values for a $v \in V$. To avoid the combinatorial state space explosion a protocol should be simplified as much as possible, while this does not compromise the verification results, of course. Over simplifications may often lead to wrong conclusions, so the amount simplifications over the protocol should be carefully applied. This is, again, the application of the abstraction paradigm [CBKK94]. However, here we are interested in the simplification of the algorithms procedures, not variable values. $\forall t \in T$, the designer should verify if t can not be reduced to a simpler representation in V, preferably as a boolean.

Finally, every time a property is verified and the tool presents a response scenario, the designer must analyze whether the result it is a fault on the protocol or on the model. For example, simplifications can introduce errors in the model in some way that some of the found properties even being present in the model are not possible in the real world. $\forall p \in P : (p \leftrightarrow G) \Rightarrow a$, the relation $\leftrightarrow$, that produces the answer a, must to be analyzed.

This is a crucial step and, unfortunately, as it is the actual technological state of the art, can not be done automatically. It is required a human operator to reasoning over the scenario and decide whether the error may or may not occur in some way.

Formal verification results

Our main aim was to verify if the protocol is loop free. Even though it seems, at first sight, loop free the protocol is not. We found out that a loop scenario may occur when the relative velocities of the nodes are not equal. For example, considering the Fig. 6.3, the simplest loop scenario occurs in the following case: node (A), acting as VAP, transmits the message $M1$ that is received by the node (B). Considering node (B) faster than node (A) and starting to act as a VAP in an ahead point in the road, it can transmit message $M1$, that if received by node (A) would characterize a loop. For this reason messages need to be equipped with unique IDs. When node (A) receives a duplicated message, identified by the ID, the node discards the message, indeed preventing the loop formation.

Yet another kind of message loop is present, and in fact it is even desirable one. Again, consider Fig. 6.3, take the node (A) acting as a VAP in the lane 1, a message $M1$ sent can reach the node (C), going in the opposite direction in the lane 2. At some point in the future the node (C) start to act as a VAP and retransmits the message $M1$ that is received by the node (D) in the lane 1. If node (D) does not have the message, it is stored and will be retransmitted in the future in case node (D) becomes a VAP. However, notice that this case is not a loop in the conventional sense, once the nodes involved are different. Other point to observe is that this kind of loop is even desirable since it helps spreading messages over the region. The

buffer favors newer messages, so when older messages will be ignored and removed from the buffer.

Even though the VAPs do not transmit when they find out there is other VAP in the same area, depending on the MAC layer protocol used, concurrent transmissions and hidden/exposed nodes problems may also occur. Here we consider the existence of a MAC layer mechanism to handle this, e.g. scheduler for IEEE 802.16 networks or CSMA/CA for IEEE 802.11 networks. However if collisions occur, then the worst impact will be a waste of bandwidth in a region that was not previously in use any way.

We also found out that may exist nodes in the network that never take advantage from the VAPs technique. There is no guarantee the mobile nodes will receive all the messages needed to fill their buffers, or a node traverses the entire path from one AP to other without receiving any message from other VAPs. This will happen if the node is unfortunate enough so as to not be inside the VAP range of other nodes acting as VAP, or when the node itself is acting as VAP for others, and thus is not receiving messages from other VAPs. These situations are more likely to occur in sparse networks.

6.4 VAPs for Disaster Scenarios

The main idea behind the Virtual Access Point concept [FCF+08] is to have mobile nodes acting as Access Points disseminating previously received messages extending the network coverage to uncovered areas. When a mobile node receives a message it stores this message in a cache. When the node perceives to be in a region without network coverage it rebroadcasts this message as if it was a regular AP. For all practical purposes, from the neighbor nodes point of view, there is no distinction between the messages received from either an AP or a VAP. The main kinds of traffic this technique targets are the ones with no strict time requirements, for example, stream traffic that can be buffered. The Algorithm 1 presents, in high level, the behavior of the VAP algorithm.

This technique is a best effort one; there are no guarantees that all packets of a stream will reach all nodes in the network, however, utilizing the presented technique enables more messages to reach more nodes. In the case of a disaster scenario, this kind of cooperative behavior can be the only way to disseminate useful general information through the network. Thus the mobile stations work as a large distributed and cooperative cache.

6.4.1 Evaluated Disaster Scenarios

For evaluation purposes, two possible scenarios are considered. The first refers to a case where the network is damaged by natural causes while in the second case, is when the network is damaged by enemy sabotage. In this chapter, the user mobility pattern is used to simulate the movement after the catastrophic events. While this is not necessarily the expected user behavior, it helps us produce results that can be used to understand how the capacity is affected. Note that although considering generating user mobility patterns that would reflect the movement of rescue workers and other users during a catastrophe would be highly desirable, it is very difficult to evaluate any such patterns. In fact any evaluation would require comparison against real traces that seem to be unavailable. The natural disasters evaluated here are earthquake and flooding, the sabotage scenarios are power outage and network random failures, these disaster scenarios where abstracted in the simulation as follows:

- Earthquake: The network starts with all the APs and mobile nodes running perfectly. However, at some point, 80% of the existing APs are randomly damaged and excluded from the network. This abstraction permits us to evaluate the effect of the technique when a major part of the APs disappear randomly from the network without any warning.

- Flooding: The evaluated scenario is a flash flooding [Col05] one. This kind of flooding, is common in mountain regions in spring, heavy rainfall during the a tropical rainy season and in the case of and in the case of dam failures. This situation is abstracted in the simulations by the random disabling of a slice of 20%, horizontal or vertical, of the middle of the network. All the APs in this segment of the network are disabled. This intends to simulate a river crossing the city that flooded in a sudden way.

- Power outage: In this scenario, we divided the evaluated scenario in four quadrants. During the simulation one of the four quadrants is randomly chased and all APs on that quadrant are disabled. Complete blackouts are rare in developed countries, but power outages in cities are relatively common if some problem occurs in a specific power station, power line or other part of the distribution system. Commonly the effect of these failures is that part of the power grid goes down letting part of the served region without energy. Such problems could occur by accident, or in consequence of sabotage.

- Random network failure: In this scenario random network APs fail and disappear from the network during the regular network operation. The degradation of the network coverage, in this case, is gradual, in contrast to what occurs in the other scenarios. This kind of generalized and chronic failure scenario

could be triggered by hacker actions or physical sabotage of the nodes to deny access to the network.

6.5 Experments

We examine two types of environments: a highway segment and a city section. The highway segment considered is 5Km long having four lanes, two in each direction with cars going back and forth on it. For the city environment we chose a $2km^2$ area of Washington DC city center, Fig. 6.4, as mapped by the Topologically Integrated Geographic Encoding and Referencing (TIGER) system with cars distributed through it. For each scenario we have 40 different configurations of 10 simulation minutes, with 200 vehicles and a transmission range of $100m$. Nodes in the city environment have minimum speed $18Km/h$ and maximum limited by each road's speed limit as registered to the U.S. Census Bureau. For the highway environment the vehicles minimum and maximum speeds are $60Km/h$ and $110Km/h$ respectively. The scenarios follow a realistic mobility pattern generated with the VanetMobiSim [HFBF06] tool.

Each generated scenario has a number of APs placed randomly. All data is presented with a five percentile and confidence interval of 99%. All simulations keep the same basic configuration and one particular parameter is varied. Variant parameters are: the stream transmission rate, the number of static APs and the

Figure 6.4: Map of the DC area considered on the experiments

method VAPs select messages to re-broadcast. The source of the stream generates CBR traffic from 1 to 3 messages per second. The number of APs tested for the city environments where 2, 25, 50 and 100. For the highway environment the number of APs evaluated where 2, 5, 10 and 15. The three ways the VAPs messages are chosen are random, oldest message first and newest message first. We used Sinalgo [Keh06], the simulation framework for testing and validating network algorithms developed in Java by the Distributed Computing Group at ETH Zurich. All the experiments were conducted using Linux Fedora Core release 6 in a Intel Xeon $1.86GHz$ machine with $16GB$ of RAM.

VAPs were first devised for highway environments. This work extends the technique and tests it in city environments. As Fig. 6.5 and Fig. 6.6 shows, it is valuable in both scenarios. Fig. 6.5 demonstrates the behavior of the VAPs for a highway environment displaying the number of unique messages received. Unique messages

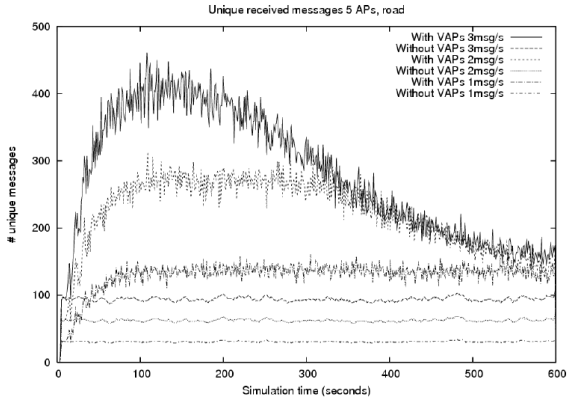

Figure 6.5: Unique received messages through the 10 minutes of simulation for the road environment with different traffic rates

are defined as messages received by a mobile node for the first time. Numbers of unique messages start to decrease around the $200s$ because at this point the caches of the nodes start saturate with stream messages and diversity of messages among the nodes caches decreases. This does not occur as much in the city environment as it does in rural environment of the simulated scenarios. In the highway scenario the cars perpetually move along the two opposite highway directions. Thus, the nodes exchange more messages but of decreased diversity. In the city environment, however, nodes follow dissimilar paths which results to diverse cache contents. The number of lost messages decreased between 10% to 15% for city environments while for the highway environment it decreased between 10% and 27.88%.

Fig. 6.7 shows the difference of having 2 or 25 APs in the city scenario for varying bit rates. Both the number of APs, and the bit rate, influences the number of unique messages received in total. However, as expected, the number of unique

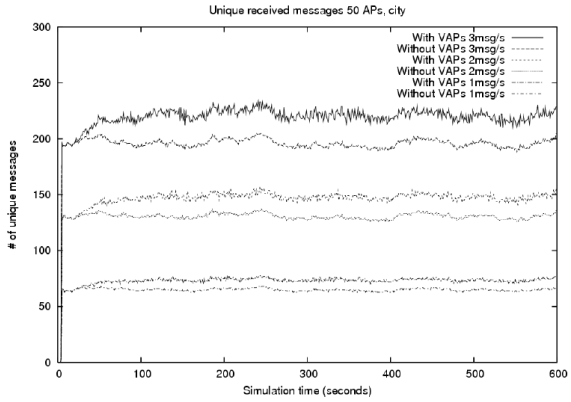

Figure 6.6: Unique received messages through the 10 minutes of simulation for the city environment with different traffic rates

messages for scenarios where VAPs are not present is nearly constant, as it only depends on the nodes passing near the APs. Even when the bit rate increases we do not observe a significant increase in the number of unique received messages. When bit rates are increased from 1 to 3 packets per second, in the 2APs case, the result is marginal. When VAPs are enabled, unique messages received significantly increase, because 2 antennas are not enough to spread the information through the entire network. The VAPs take advantage of nodes caches to propagate messages which were previously lost.

However, the bigger the covered area the lower is the gain the VAP technique presents. This becomes apparent when we look to the graph of Fig. 6.8. The graph shows, in the same experiments, the messages first received through APs and VAPs. As the number of APs nodes increases in the highway environment, the number of messages first received through VAPs decreases. The behavior is similar for the city

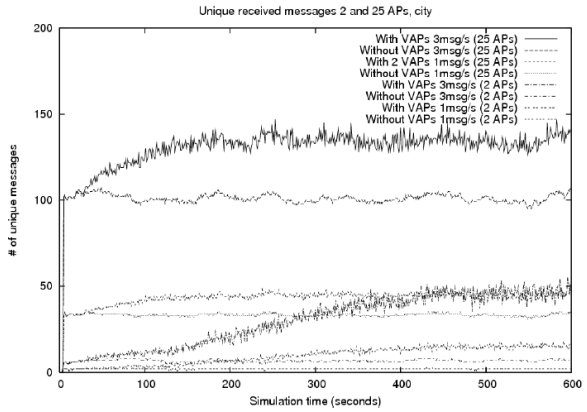

Figure 6.7: Unique received messages through the 10 minutes of simulation for the road environment with different number of APs and traffic rates

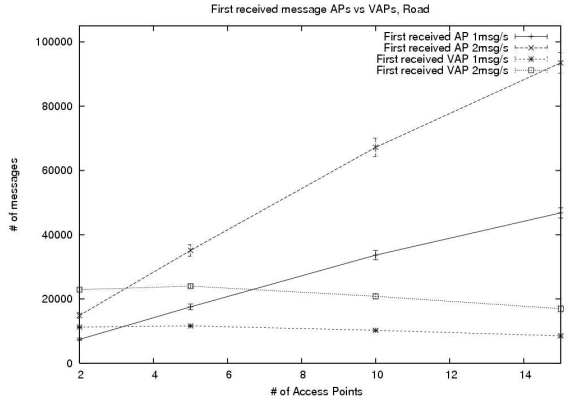

Figure 6.8: Number of messages first received from an AP and VAP

environment.

The use of VAPs accounts for an increase between 61.7%and 134.57% on the total traffic of the network. However, since this increase occurs only in non covered areas, it is not creating interference or delaying the system's APs. Nevertheless, evaluating the number of repeated messages is interesting. On, the number of repeated messages for the networks that use VAPs and the ones that do not use it

follows the same shape. Increasing the number of messages generated by the VAPs, result in an increasing of repeated messages. As depicted on Fig. 6.9 present the results for different stream rates, and also presents different transmission rates for the VAPs. Each VAP node can either transmit at the same rate the stream is generated or 4 times this rate. For example, if the stream is generated at the rate of 1 message/second (m/s), the VAP can transmit cache messages either at $1m/s$ or $4m/s$. The number of repeated messages increases based on the number of VAPs, but as the VAPs assignment is dynamic it decreases when the network coverage increases. This way the number of repeated messages also decreases, as there are less VAPs active. Ten is nearly the best number of APs for this scenario. Given less than 10 nodes, we have lot of uncovered areas and more than that the network get so over provisioned that one AP start to interfere with other and the number of repeated messages increase again, not because the VAPs, but because one mobile nodes start to receive messages from more than one AP.

Regarding the VAPs spreading messages polices, random, older to newer and newer to older, all three of them presented nearly the same results. However, on average, the random police, i.e. the VAP node sending a random message from the cache, performed slightly better than the others.

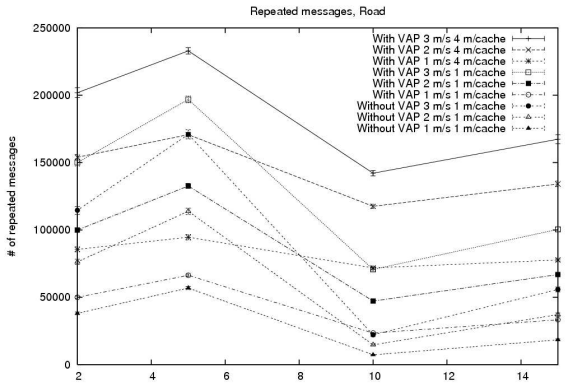

Figure 6.9: Repeated messages for the road environment

6.5.1 VAPs to Increase Network Survivability

The simulations designed to understand the impact of the VAPs on a disaster scenario area follow closely the parameters indicated for the city environment. We used the same Washington DC area, Fig. 6.4, with the same basic configuration. All experiments keep the same basic configuration but the number of and locations APs is random. Per average we allocate 40 APs but the number varies up to 100. The graphs are presented with a confidence interval of 99% and each point is the result of the mean of 34 runs with different network configurations for a period of 30 simulation minutes. The source of the stream generates CBR traffic of 1 message per second and distributed simultaneously by all the available APs. We vary the number of APs, size of the cache, disaster scenario and time, during the simulation, when the disaster occurred.

Fig. 6.10 presents the received messages map for the earthquake scenario, with

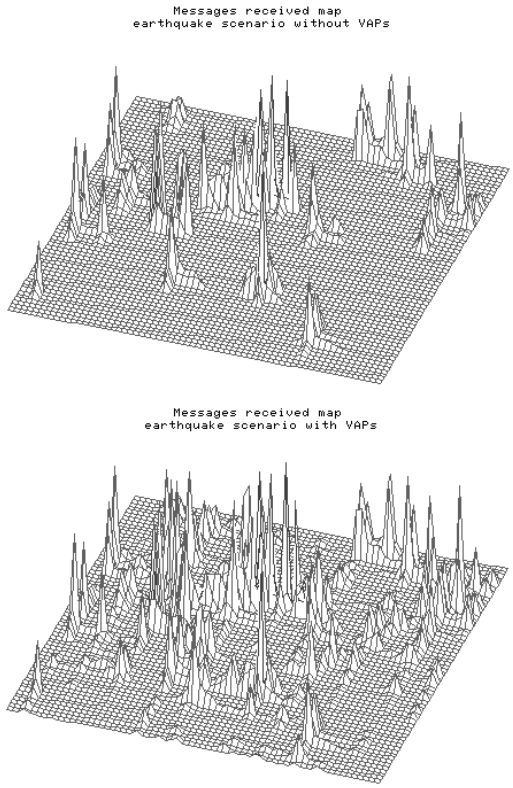

Figure 6.10: Map showing the new messages received through the simulated area

and without the use of VAPs at the considered area. Through these plots we can perceive that the use of the VAP technique not only increases the amount of received messages, but also spreads the receiving messages points through the observed area. If we compare the plotted map with the actual area map, presented in Fig. 6.4, we can even devise the roads and main intersections from it.

Fig. 6.11 shows the influence of the initial number of APs in the network and the percentage of messages received. The values represented in this graph refer to the disaster occurring in the beginning of the simulation. We can observe that for

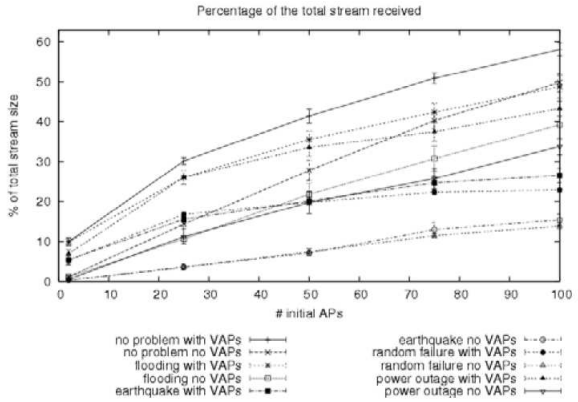

Figure 6.11: Average percentage of messages received in the network as a function of the initial number of APs for the evaluated disaster scenarios

all cases the VAP technique provides an increase in the number of stream messages received. The percentage of the stream traffic received is affected by the initial number of APs in the network, with larger number of APs, causing more extensive spread of information in the network. This makes the VAPs efficient for local traffic dissemination. In the best case, when no failure occurred in the network and all nodes work perfectly, using VAP technique provides an increase in the number of received messages that ranges from more than 700%, when the number of APs is two, to 16.6% when the initial number of access point is 100. Note that this ratio is caused by the fact that VAPs allow us to maintain communications in cases where otherwise the system would collapse.

One hundred access points represent, on average, a coverage of 58% of the total simulated area. As expected, other gain on using VAPs decreases as the space covered by access points increases. This occurs because the VAPs are well

behaved and, as it is an opportunistic protocol, the nodes act as VAP only when they are outside the range of any AP and any other VAP. With the increase of the network coverage by the real APs, the regions where a node could act as a VAP decrease and, therefore, the number of messages received through the VAPs decrease. For the disaster scenarios we can detect the same general behavior. Consequently, the percentage of the received stream is larger when the initial number of nodes increases. However, the proportional gain introduced by the VAPs decreases. For the earthquake scenario the gain varies from 1615% to 71%. In this scenario 80% of the network is damaged in the beginning of the simulation, what explains the enormous gain. In this scenario, the number of actual APs is really small and almost all the delivered messages are done through VAPs. We call gain the percent of traffic delivered with help of VAPs over the amount initially delivered without the use of the technique. For example, if we double the number of delivered messages we say the gain attributed to the VAP technique is 100%. For the flooding scenario the gain vary from 753% to 24%. In the power outage scenario case, the gain varies between 1122% and 28%. As we can see, the gain is consistent to the fraction of the initial network affected by the disaster, in the flooding scenario 20% of the network is damaged and for the power outage one fourth of the network is affected. The larger the damage in the network, the more relevant becomes the traffic received through VAPs.

For the random network failure scenario, the intervals between failures are random, distributed uniformly throughout the simulation time. By the end of the simulation only a few nodes remain functional. The damage for this scenario is not huge at first, as it happens in the earthquake scenario. However the damage is constant through time. In this way, by the end of the simulation, the damage caused to the network is comparable to the earthquake scenario. Fig. 6.15 shows the behavior of the random failure and earth quake as a function of time. For the random failure, the gain varies from approximately 1600% to 65%, close to the values estimated in the earthquake set-up. We additionally vary the buffer size and the time the disaster occurred in the simulation. The results are basically equivalent; the only difference is a small increase in the total number of received messages, when we delay the disaster start time.

The graph in Fig. 6.12 shows the number of duplicated messages received by the nodes during the experiments as result of the application of the VAP technique in function of the disaster start time. Fig. 6.13 shows the number of duplicated messages in function of the size of the cache. The values for both graphics are relatively stable. This means that the duplicate messages have a low correlation with respect to the size of the cache and the time the disasters started. As we can see in the graph in Fig. 6.12 the biggest VAP overhead is around 32% of the total number of messages sent in the stream. However, on average 10% of the duplication

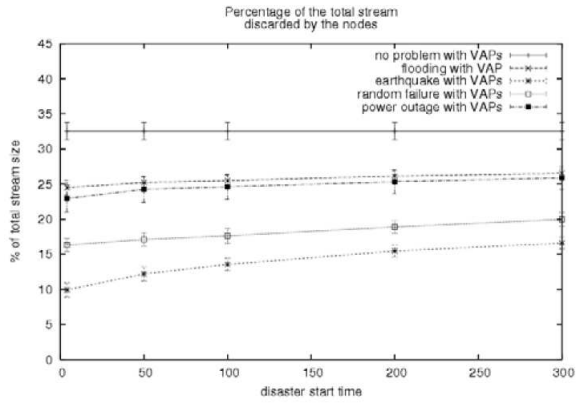

Figure 6.12: Number of duplicated messages received as a function of the initial time of each disaster

result from nodes receiving duplicated messages from APs. So the real overhead caused, in the worst case, for the VAPs is about 22% of the network stream traffic generated. When there is no disaster, all 100 APs are working without any problem and a larger part of the stream is received by the mobile nodes from the antennas. However, for the disaster scenarios, where the APs are not fully injecting traffic in the network, the overhead varies between 9% and 16% for the flooding scenario, 16% and 19% for the random failure scenario, 24% to 26% for the flooding scenario and between 22% and 25% for the power outage scenario.

In Fig. 6.14 we can observe that the number of messages received per cycle increases when we use VAPs. Using VAPs, the variability on that range increases. This should be expected since VAPs is a best effort mechanism, and not as effective as APs would have been had they been available.

Fig. 6.15 shows the number of unique messages received through time. We can

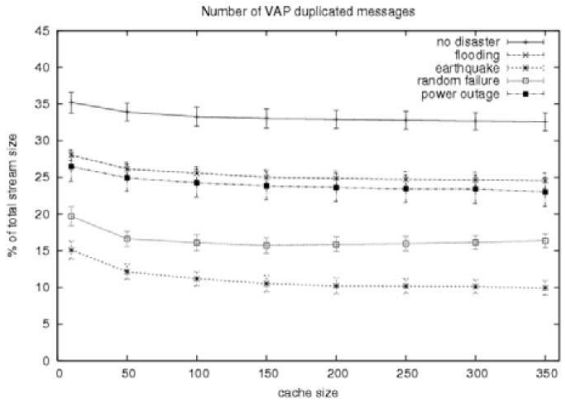

Figure 6.13: Number of duplicated messages as a function of the size of the cache with 100 nodes and the disasters occurring in the begging of the simulation

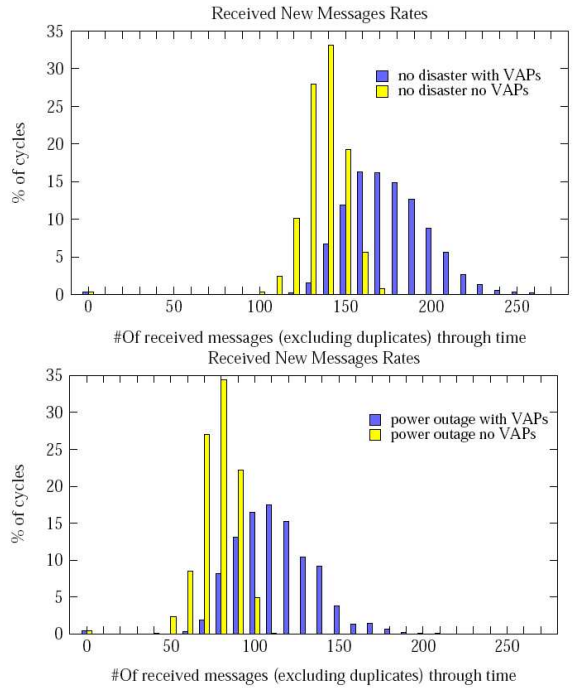

Figure 6.14: Transmission rate and variability

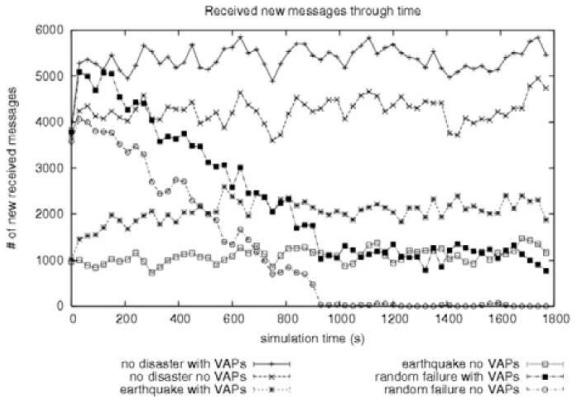

Figure 6.15: Received unique messages through time comparing the no disaster, earthquake and random failure scenarios

recognize that the use of VAPs increases the number of unique messages consistently through the time. In this graph it is interesting to notice the behavior of the random failure scenario compared to the no disaster and earthquake ones. In the beginning of the simulation the random failure and the no failure results closely resemble one another. However, as time passes, the network degrades consistently in the case of the random failure scenario. VAPs decrease the impact of the APs failures, and enables mobile nodes to receive new messages even when virtually no mobile node receives message directly from the APs. During the simulations it is guaranteed that, for any scenario, at least one AP exists and broadcasts new messages. If no VAP existed, only nodes in range of this AP would receive these new messages. Using the VAP technique these few nodes may spread the new message though the network.

6.6 Conclusion and Future work

We presented the architecture of Virtual Access Points for retransmiting streams through wireless networks. The use of this technique improves the performance of stream traffic for wireless mobile networks. It is a simple yet effective method to increase network coverage and spread stream messages in VANET networks. The technique may increase the packet reception rates, for typical scenarios, by around 30% with effectively no overhead on the APs and completely transparent to the other mobile nodes.

VAPs were also showed here as an efficient and cheap way to improve the network resilience in case of disasters. For the disaster scenarios case the increase in coverage is even more significant, which is justified since the system manages to remain operational after the initial system has collapsed. The experiments show that the gain in the number of received messages may vary from approximately 1600% to 24% depending on the disaster scenario evaluated.

The gain resulting from the application of the VAP technique in the regular network scenario varies from 755% to 16%, depending on the number of APs disseminating data in the network. Since VAPs are only used on uncovered areas, the gain observed is negatively correlated to the AP coverage of an area. As a result of applying the VAP technique, the number of duplicated and irrelevant messages received by the nodes is increased.

However, this traffic occurs in uncovered areas where it causes very low if at all interference, and in the worst case, average traffic overhead is increased by approximately 27% in the experiments discussed. When, as in our case, the nodes are mounted to vehicles, energy efficiency is not critical. However, it would be interesting to consider as an extension to this work how to increase energy efficiency when the nodes are carried by humans.

Future steps for this work include determining the optimal number of VAPs in varying environments. We also plan to implement the VAP technique on a testbed to evaluate the impact of the VAP in a real environment. In all presented results, the traffic was push based. When considering pull-based traffic, it is much more difficult to offer the level of transparency of VAPs, but one of our future goals is to extend the VAP concept to handle pullbased traffic.

Chapter 7

Conclusion

7.1 Contributions

Throughout this dissertation we demonstrated the importance of considering the effects of wireless user mobility. We considered specific instances of vehicular networks. We demonstrated our points and the results of our proposals on these systems. By applying the principles discussed on MobileCache, VAPs and PEGASUS, we demonstrated how the principles in discussion relate to each specific system.

Both PEGASUS and VAPs possess capabilities no similar system did at the time they were published. The implementation of PEGASUS lead us to finding bugs in Click's DHCP module and the port of part of Click to Android.

With MobileCache we provided a model and discussed issues for efficient data delivery to mobile users using available fixed infrastructure. We segmented queries to take advantage of multiple queries about same path segments, and developed an efficient caching mechanism. At this highly mobile setting, analysis and optimization goals discussed in past papers were irrelevant, so we provided a new analysis,

and used an adaptive scheme to efficiently adjust to changes in the distribution of requests. Mobile-Cache can efficiently answer multiple queries over a wireless environment and discuss its results. As our results show, it can sustain the same loss ratio for queries using four to five times less bandwidth than simple multicast. Additionally, as our solution is closer to the schemes when no infrastructure is present, it is more suitable where no road-side equipment is present. Our hybrid push-pull constitutes a very effective and scalable solution because it was designed taking into account user mobility.

PEGASUS was the first solution addressing a complete array of the challenges in vehicular WLAN communications. It provides wireless connection roaming at high velocities transparent to user level applications, and does not impose additional requirements to existing infrastructures. PEGASUS offers simple deployment, improved scalability, and to the best of our knowledge was the first able to operate over "in situ" secure networks. It remains efficient under intermittent connectivity conditions and supports heterogeneous network mediums for increased robustness.

It provides mobile clients with a constant IP address to preserve application sessions. Furthermore it is able to sustain connectivity in the absence of available APs by using multiple physical interfaces. Efficient connection switching is achieved by storing a global DHCP connection to the PEGASUS server and predicting connection candidates on the client's path. A salient feature of the PEGASUS system

is that it does not impose modifications to the infrastructure of deployed networks or protocols. Using in-situ infrastructure and inexpensive dynamic population of the cache helps bootstrapping the service at very low cost. Our experiments showed solid transfer rates and continuous connectivity for high velocity client simulations. The DHCP cache proved to sustain client connection transitions when the conventional connection renewal schemes degraded beyond workable conditions. We were able to achieve usable and stable network with speeds of up to 100 km/h. User connectivity can be enhanced by taking into account user mobility. For a large simulation for 200 users on the D.C. Washington area using realistic mobility patterns, we show that per average the system can sustain rates of over 5 MBps for over 87% of the time. Thus Pegasus, can provide efficient mobility support on environments of small connectivity range.

On VAPs we see how user location data can be used to optimize a delay tolerant, ad-hoc network. A new technique to allow data dissemination among vehicles was introduced. We used it to extend the reach of roadside access points to uncovered road areas. We used user mobility data to optimize the locations of VAPs and decide which set of vehicles have the potential to become effective VAPs based on their current direction of movement. Each vehicle that receives a message from an Access Point (AP) stores this message and rebroadcasts it into non covered areas. We show how this design can help us avoid interference since each

operates on a bounded region outside any AP. We see how mobility data can allow us to improve the effectiveness of VAPs when used for divert applications such as stream based traffic and broadcasting important traffic in disaster scenarios where VAPs allow the system to remain operational under conditions it would otherwise be inadequate. Virtual Access Points improve the performance of stream traffic for wireless mobile networks. It is a simple yet effective method to increase network coverage and spread stream messages in VANET networks. The technique may increase the packet reception rates by around 30% with effectively no overhead on the APs. For disaster scenarios, the experiments show that the gain in the number of received messages may vary from approximately 1600% to 24% depending on the disaster scenario evaluated. Moreover, VAP is a valuable technique to disseminate network traffic even when no disaster has occurred and can operate transparently to the system. The gain resulting from the application of the VAP technique in the regular network scenario varies from 755% to 16%, depending on the number of APs disseminating data in the network. Since VAPs are only used on uncovered areas, the gain observed is negatively correlated to the AP coverage of an area. As a result of applying the VAP technique, the number of duplicated and irrelevant messages received by the nodes is increased. However, this traffic occurs in uncovered areas where it causes very low if at all interference, and in the worst case, average traffic overhead is increased by approximately 27% in the experiments discussed.

BIBLIOGRAPHY

[AAFZ95] Swarup Acharya, Rafael Alonso, Michael Franklin, and Stanley Zdonik. Broadcast disks: data management for asymmetric communication environments. In ACM SIGMOD 1995, pages 199–210, 1995.

[ABB+04] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications, pages 121–132, New York, NY, USA, 2004. ACM Press.

[AFZ97] Swarup Acharya, Michael J. Franklin, and Stanley B. Zdonik. Balancing push and pull for data broadcast. In ACM SIGMOD, pages 183–194, 1997.

[BB95] Ajay Bakre and B. R. Badrinath. I-tcp: indirect tcp for mobile hosts. In Proceedings - International Conference on Distributed Computing Systems, page 136, Vancouver, Can, 1995. IEEE, Piscataway, NJ, USA. r10.

[BCC98] Sergey Berezin, Srgio Campos, and Edmund M. Clarke. Compositional reasoning in model checking, 1998.

[BGJL06] John Burgess, Brian Gallagher, David Jensen, and Brian N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking. In Proceedings of IEEE Infocom 2006, Barcelona, Spain, April 2006.

[BH00] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks, 2000. 7.

[BHM+06] Vladimir Bychkovsky, Bret Hull, Allen Miu, Hari Balakrishnan, and Samuel Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In Mario Gerla, Chiara Petrioli, and Ramachandran Ramjee, editors, MOBICOM, pages 50–61, Los Angeles, CA, September 2006. ACM. bychkovsky2006measurement,r5.

[Bro08] Gabriel Brown. LTE Base Stations and the Evolved Radio Access Network. Heavy Reading Reports, 6(16), December 2008.

[BSAK95] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving tcp/ip performance over wireless networks. Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 2–11, 1995. r11.

[BTAZ06] Muhammad Mukarram Bin Tariq, Mostafa Ammar, and Ellen Zegura. Message ferry route design for sparse ad hoc networks with mobile nodes. In Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '06, pages 37–48, New York, NY, USA, 2006. ACM.

[BWLW04] Bharat Bhargava, Xiaoxin Wu, Yi Lu, and Weichao Wang. Integrating heterogeneous wireless technologies: a cellular aided mobile ad hoc network (cama). Mob. Netw. Appl., 9(4):393–408, 2004. r13.

[CBKK94] Peter J. Clarke, Djuradj Babich, Tariq M. King, and B. M. Golam Kibria. Model checking and abstraction. ACM Transactions on Programming Languages and Systems, 16:1512–1542, 1994. 31.

[cel10] 3G Network Comparison:AT&T, Cricket, Metro PCS, Sprint, T-Mobile, US Cellular & Verizon . `http://www.cellularmaps.com/3g_compare.shtml/`, 2010. [Online; accessed August-2011].

[CFF+08] Daniel Câmara, Nikolaos Frangiadakis, Fethi Filali, A. A. F. Loureiro, and Nick Roussopoulos. Virtual access points for stream based traffic dissemination. In APSCC 2008, 3rd IEEE Asia-Pacific Services Computing Conference, December 9-12, 2008, Yilan, Taiwan, 12 2008.

[CFF+09] Daniel Câmara, Nikolaos Frangiadakis, Fethi Filali, Antonio A F Loureiro, and Nick Roussopoulos. Virtual access points for disaster scenarios. In WCNC 2009, IEEE Wireless Communications & Networking Conference, April 5-8, 2009, Budapest, Hungary, 04 2009.

[CKV01] Z. Chen, H. Kung, and D. Vlah. Ad hoc relay wireless networks over moving vehicles on highways, 2001.

[CLF07] Daniel Câmara, Antonio Alfredo F Loureiro, and Fethi Filali. Methodology for formal verification of routing protocols for ad hoc wireless networks. In GLOBECOM 2007, 50th IEEE Global Communications Conference, November 26-30, 2007, Washington, USA, 11 2007.

[Col05] Larry Collins. Technical Rescue Operations: Common Emergencies. PennWell Books, 2005.

[DMJR05] R. Dilmaghani, B. Manoj, B. Jafarian, and R. Rao. Performance evaluation of rescuemesh: a metro-scale hybrid wireless network. In WiMesh 2005: First IEEE Workshop on Wireless Mesh Networks held in conjunction with SECON, Santa Clara, Sep. 2005.

[DNP+04] S. Das, A. Nandan, G. Pau, M.Y. Sanadidi, and M. Gerla. SPAWN: Swarming Protocols for Vehicular Ad Hoc Wireless Networks. In VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, pages 1–9, New York, NY, USA, 2004. ACM Press. 1.

[EJs02] M. Esbjrnsson, O. Juhlin, and M. stergren. The hocman prototype: Fast motor bikers and ad hoc networks, 2002. r9.

[EP05] E. Efstathiou and G. Polyzos. Self-organized peering of wireless lan hotspots. In European Transactions on Telecomunications, vol. 16, no. 5 (Special Issue on Self-Organization in Mobile Networking), Sept/Oct 2005.

[FCF+08] Nikolaos Frangiadakis, Daniel Câmara, Fethi Filali, Antonio Alfredo F Loureiro, and Nick Roussopoulos. Virtual access points for vehicular networks. In Mobilware 2008, 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, February 12th-15th, 2008, Innsbruck, Austria, 02 2008.

[FVDN04] Qiu Fang, Susan V. Vrbsky, Yu Dang, and Weigang Ni. A pull-based broadcast algorithm that considers timing constraints. In ICPPW 04, pages 46–53, Washington, DC, USA, 2004. IEEE Computer Society.

[FZ98] Michael J. Franklin and Stanley B. Zdonik. "data in your face": Push technology in perspective. In Laura M. Haas and Ashutosh Tiwary, editors, ACM SIGMOD 1998, pages 516–519, 1998.

[Gav01] Jr. Gavrilovich, C.D. Broadband communication on the highways of tomorrow. Communications Magazine, IEEE, 39(4):146 –154, April 2001.

[GBMY97] D.J. Goodman, J. Borras, N.B. Mandayam, and R.D. Yates. Infostations: a new system model for data and messaging services. In IEEE 47th Vehicular Technology Conference, Phoenix, AZ, USA, 1997.

[GL94] Orna Grumberg and David E. Long. Model checking and modular verification. ACM Trans. Program. Lang. Syst., 16:843–871, May 1994.

[GSD06] Richard Gass, James Scott, and Christophe Diot. Measurements of in-motion 802.11 networking. In WMCSA '06: Proceedings of the Seventh IEEE Workshop on Mobile Computing Systems & Applications, pages 69–74, Washington, DC, USA, 2006. IEEE Computer Society. r8.

[HBZ+06] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems, pages 125–138, New York, NY, USA, 2006. ACM Press.

[HFBF06] J. Harri, F. Filali, C. Bonnet, and Marco Fiore. Vanetmobisim: generating realistic mobility patterns for vanets. In VANET 06, pages 96–97, New York, NY, USA, 2006. ACM Press.

[JHP+03] Jorjeta G. Jetcheva, Yih-Chun Hu, Santashil PalChaudhuri, Amit Kumar Saha, and David B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. wmcsa, 00:32, 2003.

[Keh06] Roger Kehrer. ManS A Simulation Framework for Mobile Ad-Hoc Networks. PhD thesis, ManS A Simulation Framework for Mobile Ad-Hoc Networks, 2006.

[KFHM04] Miltiadis Kyriakakos, Nikolaos Frangiadakis, Stathes Hadjiefthymiades, and Lazaros F. Merakos. Rmpg: a realistic mobility pattern generator for the performance assessment of mobility functions. Simulation Modelling Practice and Theory, 12(1):1–13, 2004.

[KHA] David Kotz, Tristan Henderson, and Ilya Abyzov. CRAWDAD data set dartmouth/campus.

[KM04a] Dimitrios Katsaros and Yannis Manolopoulos. Broadcast program generation for webcasting. Data Knowl. Eng., 49(1):1–21, 2004.

[KM04b] Dimitrios Katsaros and Yannis Manolopoulos. Web caching in broadcast mobile wireless environments. IEEE Internet Computing, 08(3):37–45, 2004. 13.

[KR06] P. Klapwijk and L. Rothkrantz. Topology based infrastructure for crisis situations. In Proceedings of the 3rd International ISCRAM Conference, May 2006.

[LR00] Qun Li and Daniela Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In Mobile Computing and Networking, pages 44–55, 2000.

[LUKY96] Henry Lorin, Helena Unger, Per Kulling, and Lars Ytterborn. The great hanshin-awaji (kobe) earthquake january 17. KAMEDO Report, 66:12, 1996.

[MC02] McKinsey and Co. Increasing fdnys preparedness. New York City Fire Department web site, 2002.

[MKJK99] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The click modular router. In SOSP '99: Proceedings of the seventeenth ACM symposium on Operating systems principles, pages 217–231, New York, NY, USA, 1999. ACM Press.

[NBG06] Valery Naumov, Rainer Baumann, and Thomas Gross. An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces. In MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing, pages 108–119, New York, NY, USA, 2006. ACM Press.

[NDP+05] Alok Nandan, Shirshanka Das, Giovanni Pau, Mario Gerla, and M. Y. Sanadidi. Co-operative downloading in vehicular ad-hoc wireless networks. In WONS '05: Proceedings of the Second Annual Conference on Wireless On-demand Network Systems and Services (WONS'05), pages 32–41, Washington, DC, USA, 2005. IEEE Computer Society. 2.

[OK] Jorg Ott and Dirk Kutscher. A disconnection-tolerant transport for drive-thru internet environments. In INFOCOM 2005. Twenty-fourth

AnnualJoint Conference of the IEEE Computer and Communications Societies, pages 1849–1862.

[OK04a] J. Ott and D. Kutscher. The "drive-thru" architecture: Wlan-based internet access on the road. In Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th, pages 2615–2622, New York, NY, USA, 2004. ACM Press.

[OK04b] J. Ott and D. Kutscher. Drive-thru internet: Ieee 802.11b for "automobile" users. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, volume 1, 2004.

[oTAUtUS04] National Commission on Terrorist Attacks Upon the United States. The 9/11 commission report: final report of the nationa commission on terrorist attacks upon the united states. Technical report, National Commission on Terrorist Attacks, 2004.

[Pnu85] A. Pnueli. In transition from global to modular temporal reasoning about programs, pages 123–144. Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[RCC+04] Pablo Rodriguez, Rajiv Chakravorty, Julian Chesterfield, Ian Pratt, and Suman Banerjee. Mar: a commuter router infrastructure for the mobile internet. In MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services, pages 217–230, New York, NY, USA, 2004. ACM Press. r12.

[Res11] ABI Research. Wi-fi equipment market data. Technical Report 1003910, ABI Research, New York, USA, November 2011.

[SRB97] Konstantinos Stathatos, Nick Roussopoulos, and John S. Baras. Adaptive data broadcast in hybrid networks. In The VLDB Journal, pages 326–335, 1997.

[SRJB03] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks, 2003.

[ST00] Chi-Jiun Su and Leandros Tassiulas. Joint broadcast scheduling and user's cache management for efficient information delivery. Wirel. Netw., 6(4):279–288, 2000.

[STS+07] Nishant Sinha, Don Thomas, Dawn Song, Corina Psreanu, and Oded Maler. Automated compositional analysis for checking component substitutability, 2007.

[TM05] Anthony M. Townsend and Mitchell L. Moss. Telecommunications infrastructure in disasters: Preparing cities for crisis communications. Technical report, Center for Catastrophe Preparedness and Response & Robert F. Wagner Graduate School of Public Service, New York University, May 2005.

[TN06] Liviu Iftode Tamer Nadeem, Pravin Shankar. A comparative study of data dissemination models for vanets. In Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS 2006), July 2006.

[Tou] Jean Tourrilhes. Wireless extensions for linux. Wireless Tools for Linux.

[URL] URL. Homepage of fleetnet project. http://www.fleetnet.de/.

[wif10] WiMAX and WiFi Together: Deployment Models and User Scenarios. `http://www.motorola.com/web/Business/Solutions/`, 2010.

[XRDD03] Qing Xu, Segupta R., Jiang D., and Chrysler D. Design and analysis of highway safety communication protocol in 5.9 ghz dedicated short range communication spectrum. In Vehicular Technology Conference, 2003. VTC 2003-Spring. The 57th IEEE Semiannual, 2003.