UNIVERSITY OF CALIFORNIA

Los Angeles

Mobile Service Continuity in a Heterogeneous Wireless Network Environment

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Jorge Mena

2017

ABSTRACT OF THE DISSERTATION

Mobile Service Continuity in a Heterogeneous Wireless Network Environment

by

Jorge Mena
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2017
Professor Mario Gerla, Chair

Mobile devices such as laptops and smartphones are currently equipped with multiple network interfaces, allowing them to reach the Internet via multiple paths. When a running application connects to a service in a remote computer host through the Internet, it leverages on transport and network protocols such as those in the TCP/IP stack expecting continuous, fluent connectivity regardless of which network the mobile device uses. The current state-of-the-art transport protocol that dominates in the Internet is TCP; however, TCP does not allow continuous connectivity when a mobile device leaves the network coverage of an access point. It is expected from a typical device that can connect to both WiFi and cellular networks that it can automatically hand over to an available cellular network once WiFi coverage disappears, and vice-versa. Multipath TCP is a recently proposed transport protocol, backward compatible to TCP, that allows for seamless handovers when new Internet paths appear. This work studies the performance of MPTCP to truly achieve service continuity, with handovers whenever new Internet paths are available, in highly mobile scenarios, such as vehicles engaged into VANET configurations. There may be situations, however, when handovers lead to performance degradation because the newly discovered Internet paths have poor network characteristics. This thesis also addresses the problem of path selection in MPTCP when mobile devices are configured to use at most one network interface prioritizing battery performance; this is the default configuration in smartphones. In all, this thesis advocates for the use of Multipath TCP to truly achieve service continuity for end-to-end connections in both static and highly dynamic mobile settings. It does not consider MPTCP as a substitution to TCP but a natural evolution from the Internet it was designed for to Internet today.

The dissertation of Jorge Mena is approved.

Levon Goukasian

Lixia Zhang

Alfonso F. Cárdenas

Mario Gerla, Committee Chair

University of California, Los Angeles

2017

To my parents Saúl and María Victoria and my brother Saúl. It's my turn.

Para mis padres Saúl y María Victoria y mi hermano Saúl. Ahora sigo yo.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

My eternal gratitude to my adviser Mario Gerla, for his endless patience and invaluable guidance; his great charisma and personality help me develop both professional and personally, and for that I am very thankful.

Thanks to all my fellow NRL lab researchers with whom I spend endless hours on research topic discussions. To the UCLA CS Department staff who were always there to attend and help me with my petty requests. Special thanks to all the Master's Degree students that approached to me looking for academic advise and research topics; your success makes me so proud. Many thanks also to Joey Chou, from Intel Corporation, for his leadership and friendship during my internship time.

And very special thanks to my girlfriend Katherine for her understanding and support during my long days and nights in my PhD life. Thank you, Katelina!

VITA

2000–2004	Bachelor’s Degree in Computer Science and Information Systems, Department of Computer Science, University of California, Riverside
Summer 2006	Operating Systems Instructor, Department of Computer Science, University of California, Riverside
Summer 2007	Operating Systems Instructor, Department of Computer Science, University of California, Riverside
2004–2007	Teacher Assistant, Department of Computer Science, University of California, Riverside
2004–2007	Master’s Degree in Computer Science, Department of Computer Science, University of California, Riverside
2012	Wireless Systems Intern, Intel Corporation
2014	Software Engineer Intern, IBM Corporation
2011-2017	Teacher Assistant and Graduate Student Researcher, Department of Computer Science, University of California, Los Angeles
2015–2016	Head Teacher Assistant, Department of Computer Science, University of California, Los Angeles

PUBLICATIONS

J. Mena, M. Gerla, V. Kalogeraki. “Mitigate Funnel Effect in Sensor Networks with Multi-Interface Relay Nodes.” *IEEE DCOSS12*.

L. C. Tung, J. Mena, M. Gerla, C. Sommer. “A Cluster Based Architecture for Intersection Collision Avoidance Using Heterogeneous Networks.” *IEEE Med-Hoc-Net13*.

P. Du, S. Nazari, J. Mena, R. Fan, M. Gerla, R. Gupta. “Multipath TCP in SDN-enabled LEO Satellite Networks.” *IEEE MILCOM16*.

J. Mena, M. Gerla, F. F. González Navarro. “Vehicular Network Operating on a Cognitive Network.” Technical Report *UCMEXUS/UCLA 2016*.

J. Mena, P. Bankole, M. Gerla. “Multipath TCP on a VANET: A Performance Study.” *ACM SIGMETRICS17*.

J. Mena, Y. Gao, M. Gerla. “MPTCP Path Selection using CapProbe.” *IEEE WCNC18*, under review.

J. Chou, J. Mena. “Identifying Coverage Holes Using Inter-RAT Handover Measurements.” *US Patent 9660710, 8868067*.

J. Chou, J. Mena. “Coverage Adjustment in E-UTRA Networks.” *US Patent 9538413, 9154978, 8995255*.

J. Chou, J. Mena. “System and Method of Cell Outage Compensation in Cellular Systems.” *US Patent 9363727*.

CHAPTER 1

Introduction

The advancement of embedded systems and wireless communication technologies have opened the door to many applications in mobile platforms. According to the Pew Research Center [34], mobile devices such as smartphones and laptops proliferate among adults. Since 2009, the data usage on a cellphone has doubled reaching 60%; 91% of adults own a cellphone, 67% of them find themselves checking their device for updates; and 40% admit to even sleep close to them. Cisco, in its Visual Networking Index Forecast [103], has identified the following predictions by 2020:

1. Traffic from wireless and mobile devices will account for two-thirds of total IP traffic.
2. Smartphone traffic will exceed PC traffic, with 30% of the total.
3. M2M (i.e., sensors) devices traffic will grow 44% annually.
4. Tablet traffic will grow 39% annually.
5. Every second, nearly a million minutes of video content will move across the Internet.
6. Video surveillance, virtual-reality, Internet video to TV, VoD and CDN will lead the video traffic.

According to comScore, a company that measures consumer data traffic on the Internet, in its Cross-Platform Future in Focus Report 2017 [83] it reports:

1. Smartphone penetration in the US is 81% as of December 2016.
2. People spend 70% of their digital time on a mobile device.

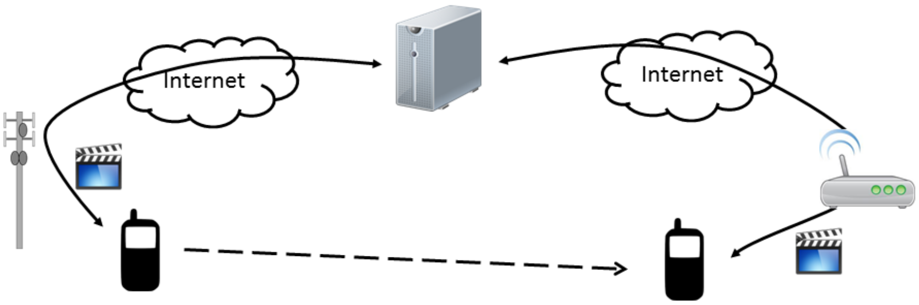


Figure 1.1: Service continuity during a RAT handover.

3. 20% of the time spent on a mobile device is on Social Media, 14% on Multimedia, and 11% on Entertainment.

These figures and trends tell us the kind of impact that mobile device technologies has and will continue having on the market; it is clear the importance that wireless technology research has in today's society. The increase in demand for mobile devices and the consumption of their content is evident. If the mobile market is to succeed, the services it offers need to connect to the outside world, i.e. the Internet. Over the years, several Radio Access Technologies (RAT), e.g. wireless, cellular, satellite networks, etc., were created to fulfill this goal. Rather than having a one network that solves all the connection problems, today we find mobile devices equipped with multiple network interfaces such as WiFi, Bluetooth, and LTE that can be used to reach the Internet. Users expect to enjoy continuous connectivity from these network technologies regardless of which one is currently in use or how the devices moves across them. We refer to this characteristic as *Service Continuity*, or the transparent operation of any network, regardless of which network technology is used for communication, Figure 1.1 shows this concept.

Mobile devices provide services to users such as voice, email, web browsing, etc., leveraging a wireless network; at the same time it needs to conserve energy as much as possible. Each network technology has its benefits and drawbacks. A WiFi network can provide high data rates at a lower battery cost and relatively inexpensive operation; on the other hand, a cellular network provides broader coverage allowing a higher degree of mobility. Both technologies seem complementary and it would be reasonably to think that they can be used to provide continuous services under

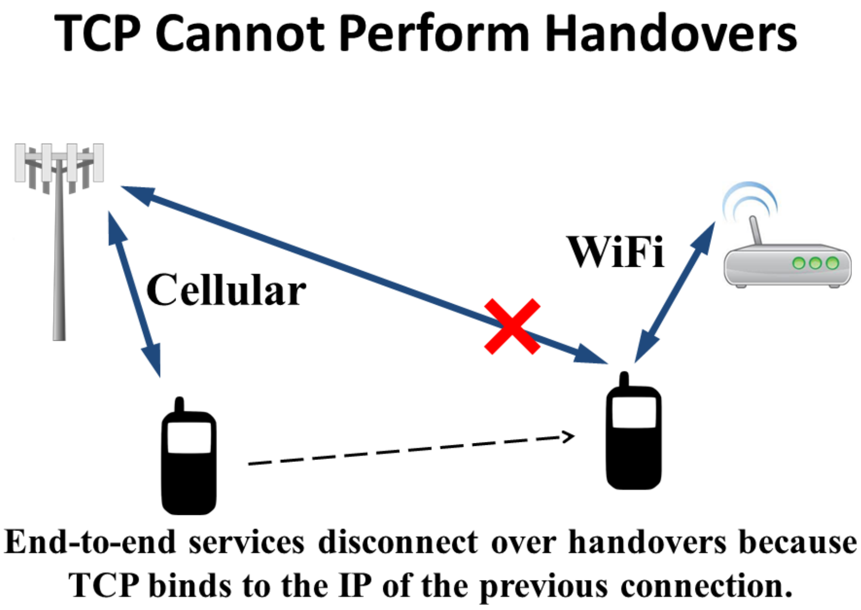

TCP Cannot Perform Handovers
Cellular
WiFi
End-to-end services disconnect over handovers because
TCP binds to the IP of the previous connection.

Figure 1.2: A TCP end-to-end session must disconnect before a handover takes place.

mobility; unfortunately, it may not be quite the case currently.

A transfer of an Internet connection from one network to another is known as a network *handover* and they should occur in a make-before-break manner, i.e. without disconnecting it. A connection performing a video transmission may be handed over from WiFi onto LTE, for example, if the user moves out of her WiFi base station's coverage. Handovers, however, break TCP connections in current configurations. TCP is the state-of-the-art transport service in the Internet that establishes end-to-end connectivity between hosts, it provides message reorder and recovery during network congestion and it carries more than 90% of the Internet's traffic nowadays; that is why they are relevant. TCP's original design does not allow seamless handovers between networks; TCP binds both the host's and remote's IP address and a port number, which makes it impossible to migrate the connection from one IP address to another during a handover without breaking the service connection; this issue is shown in Figure 1.2. This is a major problem since TCP is the dominant transport protocol currently deployed worldwide and we currently live in a highly mobile world. Currently, users rendering video on their mobile devices need to stop the video, change from WiFi to the LTE network and restart the video again; this is clearly inefficient and undesirable.

Multipath TCP, or MPTCP [44], is a recently proposed transport protocol, backward compatible to TCP that addresses the previous handover issue with TCP. MPTCP leverages the mobile devices' multiple interfaces to provide service continuity, seamlessly, to applications running on top of the network stack. In short, MPTCP uses *TCP subflows*, or independent TCP connections per network interface (WiFi, LTE, Ethernet, etc.) coupled together by a higher level engine in order to transparently provide the transport service to the application running at higher layers. Unfortunately, MPTCP inherits another well known problem from TCP: losses due to wireless lossy links.

A *wireless lossy* link in a wireless connection is a connection that experiences a significant amount of bit error rate that degrades the performance of higher layer protocols and applications. A packet carried over a lossy link that gets corrupted during its transmission is known as a *packet loss*. In TCP, packet losses are considered congestion results; therefore, TCP would unnecessarily activate its congestion control algorithms in an attempt to mitigate the event resulting in network throughput degradation. However, random losses do not indicate congestion and when they occur, TCP's performance is negatively impacted. MPTCP uses TCP for each subflow it establishes over a new wireless interface; therefore, MPTCP inherits the same issue with wireless lossy links at a per-subflow basis.

1.1 Goal of the Thesis

The principal goal of this thesis is to provide service continuity for network connections in mobile devices. The work described in this thesis leverages MPTCP to achieve such goal in the presence of new networks in a heterogeneous environment. Given the fact that MPTCP is also vulnerable to wireless lossy links, this thesis also focuses on whether a network handover should take place or not; in other words, this thesis aims to investigate if a device connection should stay in the current path or if it should go ahead with the handover. This problem is known as *path selection*; if several networks are available to a mobile device, it is fundamental to understand the path characteristics in each network before a handover occurs. We study this problem and provide solutions in several

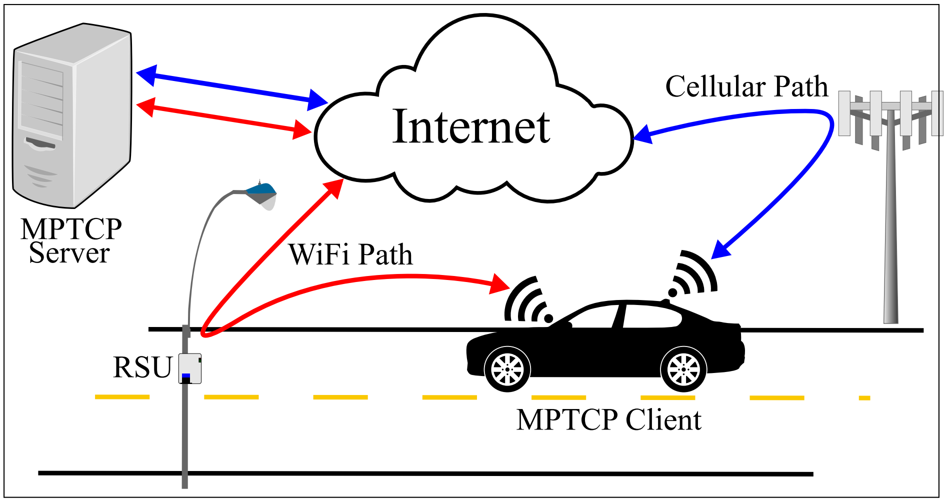


Figure 1.3: VANET Vehicle-to-Infrastructure configuration

scenarios, including pedestrian and vehicular.

1.2 Operating Scenarios

A mobile device operates in a mobile environment as shown in Figure 1.1; the devices are equipped with multiple interfaces capable of establishing connections with multiple networks. They are also equipped with a battery pack that provides energy resources for a limited time. In terms of computational power, our mobile devices may perform the complex computations than any current smartphone in the market performs; in general, these include the ability to compute complex algorithms in time and space constrains.

There are also access points that provide network connection to our mobile devices; an example of this configuration might be a VANET in vehicle-to-infrastructure configuration as shown in Figure 1.3 or simply a user walking into a WiFi hot stop while using his phone's cellular connection. It is possible that mobile devices may also establish self-configuring wireless networks or *ad-hoc networks* as well; for example, a VANET sharing cellular access using V2V connections as in Figure 1.4. Mobile devices move freely across an area covered by wireless networks using a pedestrian and vehicular mobility characteristics; they find access points and may perform handovers assuming that they are allowed by the network operators. Finally, in some scenarios we use wired networks to compare or emulate cellular networks in test beds when the use of an actual

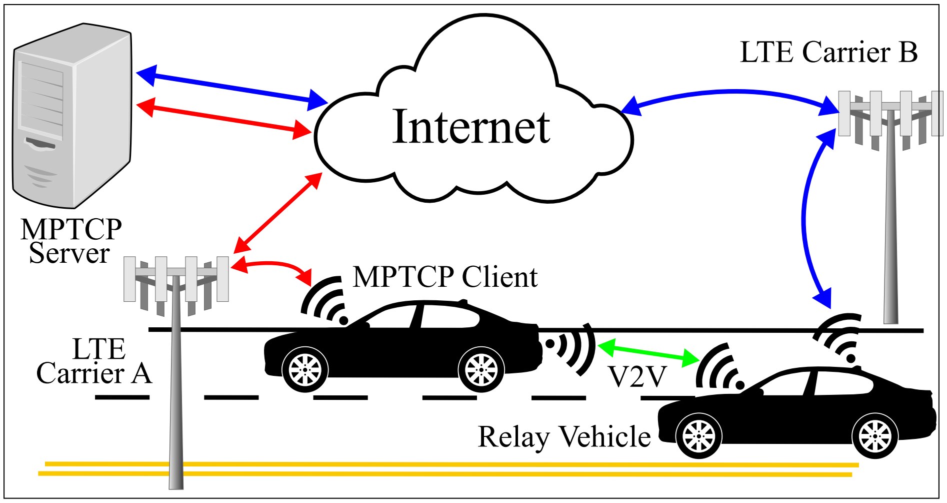


Figure 1.4: VANET Vehicle-to-Vehicle configuration.

cellular networks cost is prohibitive.

1.3 Impact of Thesis

The concept of service continuity and efficient path selection apply to a broad range of real life scenarios and the benefit of a system that operates uninterruptedly is clearly beneficial for its users. For example, a cellular network is an example of a broadband communication service; LTE, [28, 112, 78, 38], is currently the best cellular network realized by the 3GPP Consortium group. One of the major characteristics of LTE is that all communication is packet-switched; in other words, LTE traffic may be carried on the Internet; this contrasts with older technologies like GSM that used circuit-switch connections. This characteristic opens the possibility of handing voice and especially data communication flows between mobile devices on another wireless technologies, such as WiFi, if they are within proximity. Proximity Services (or ProSe), [2], is an LTE service that aims to provide location information of mobile devices in order to establish a direct communication if they are within proximity. WLAN Offloading, [3], is also an LTE service that aims to offload LTE connections onto the WLAN. In conjunction, ProSe and WLAN Offloading provide the lower layer connection handovers mechanisms that MPTCP may take advantage of to perform multipath connections and service continuity.

This thesis addresses two real scenarios where a heterogeneous network may have a positive impact in society: sensor networks and vehicular networks. In sensor network scenarios and smart grids, data collectors and aggregators may be used to alleviate potential excess in data generation (events) or communication resources (electromagnetic spectrum). In the first case, the sensor network should devote its resources to sensing rather than relaying data messages [95, 93]; in the second case, it is preferable that a data collector performs the delivery of the measurements on behalf of the sensors in a smart grid network. When fewer devices demand resources to a broadband network, a decrease in traffic requests will be experienced and hence the service will be more responsive. In both cases, a multi-radio device operating in a heterogeneous environment may act as a relay and use its multiple interfaces to deliver messages to foreign destinations. These relay devices provide service to its users (sensors) allowing them to focus on collecting sensorial data and better battery efficiency while network resources are optimized. In such a heterogeneous environment, the use of MPTCP in relays nodes would increase the lifespan of IoT and sensor networks, thus extending their utilization in areas where monitoring is necessary; for example, wild-fire forest monitoring, wireless health, surveillance, etc.; longer sensor monitoring would be achieved.

Another area addressed in this thesis where a heterogeneous network applies is on vehicular networks. In vehicular networks, mobile devices installed on cars collect traffic information and environmental data in order to improve a vehicular traffic or enhance the environment conditions. For example, vehicles may report geo-temporal information regarding their current location in order to model the traffic dynamics in a region; or vehicles may also collect pollution data using sensors in order to map regions of high concentration of pollutants and perform a network decision (perhaps reduce vehicle traffic over this area) as a preventive response. If the goal is to maximize the collection of events in their environment, we have a communication optimization issue; if the goal is to minimize traffic, we have a capacity optimization problem. In both cases, there is an inter-vehicle data exchange. Vehicles cannot naively assume that all their traffic may go through the cellular network or any other broadband network that covers them because it becomes a limited resource. It is more efficient to use several wireless technologies if possible; offloading one network onto another whenever possible. If vehicles are close to an intersection, they may use WiFi if available, or an ad-hoc network for inter-vehicle communication to offload the traffic; otherwise, broadband is used as a last resort. When traffic systems leverage on vehicles that continuously sense and share their findings and when they are able to alleviate traffic jam conditions, reduce pollution and reallocate network resources, it undeniably benefits society.

1.4 Thesis Organization

This thesis is organized as follows. This thesis first provides a broad background of several study areas relevant to this thesis in Chapter 2. Next the thesis moves to explore some VANET solutions that illustrate the use of Heterogeneous Networks in Chapter 3. In order to demonstrate the potential benefits of cellular networks and their technologies, as well as providing some solutions for coverage issues, this thesis presents the relevant patents granted during an internship at Intel Corporation in Chapter 4. This thesis moves onto the viability of Multipath TCP in wireless networks, leveraging LTE and WiFi, first on a VANET configured as a V2I in Chapter 5, and later configured as V2V in Chapter 6. The path selection problem is address in Chapter 7 with the use of CapProbe and their derived metrics. Topics on Sensor Networks, IoT, and Cognitive Networks relevant to MTPCP path selection are discussed in Chapter 8. Finally conclusion of the thesis is given in Chapter 9.

1.5 Author's Previous Work

This section presents a summary of the author's previous work completed relevant for this thesis.

1.5.1 Dynamic Relay Node Placement

In a wireless sensor network, nodes build an ad-hoc network and cooperate with the routing and relaying of data messages once they are deployed. This behavior drains the node's power resources in exchange of keeping the surveillance network connected. There exists mechanisms that control this drainage but their effect is marginal when the network experiences a high number of relevant events that trigger the nodes. Moreover, as the nodes consume their battery packs, they shut down and provoke network disconnections and segmentation. The consequences are the decrease of route

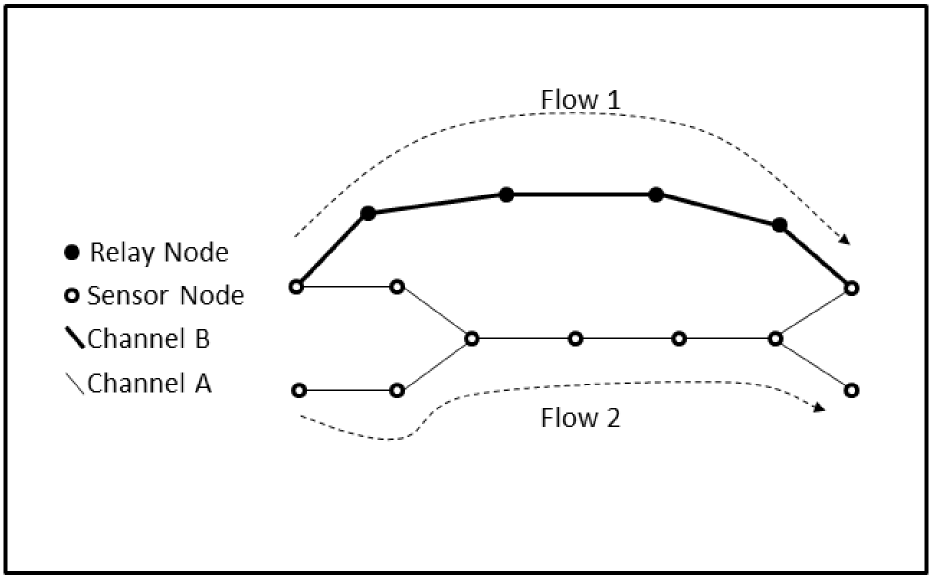


Figure 1.5: Depiction on how relay nodes may support a bottleneck topology

path alternatives towards the sink, the increase in network congestion, and the further drainage of energy resources in an accelerating rate.

The use of relay nodes, or mobile wireless nodes equipped with higher computational and energy power, takes a different approach at mitigating the high consumption of the network resources. If one or a group of wireless sensor nodes are capable of detecting the extent of the network congestion, it is possible to deploy a temporary overlay of these relay nodes in order to add more network resources in the form of new routing paths using non-overlapping frequencies so that the underlying network of sensor nodes is unaffected. The Dynamic Relay Node Placement [95] takes this approach.

An example of this approach is shown in Figure 1.5. From the figure we see a bottleneck topology that two data flows use to deliver their messages using a common frequency A. If no relay nodes are added, the nodes in the chain that carries $Flow2$ will eventually consume their battery packs relaying data messages. A set of four relay nodes R can be strategically placed so that one of the two flows gets diverted. Moreover, if these relay nodes R use an orthogonal frequency B to A, then no interference will be added by the transmission of their data messages and thus network congestion due to packet errors will decrease.

In such a heterogeneous environment, sensors leverage the best path provided by the relay nodes when such resources are available while minimizing the use of their current sensor path. In the context of this thesis, this scenario is ideal for MPTCP-enabled sensors. When sensors establish end-to-end connections, they use the sensor network to find a path; when relay nodes provide another path, MPTCP enables the sensors to add an extra path for an enhanced communication. For further details, see [95].

1.5.2 Funnel Effect Mitigation

This work addresses the specific case of network congestion in wireless networks with multiple source flows and a single destination, the *funnel effect*. Since sensor networks generate their traffic from wide, sparse areas around the destination sink, sensors that relay messages across the network experience a geometric demand of spectrum resources as they approximate the destination sink. The consequence is that sensors closer to the sink tend to consume their resources at a faster rate than those outside because they miss their duty cycles and are constantly relaying data, rather than sensing. The work introduced in Section 1.5.1 addressed the issues with bottlenecks created due to resource exhaustion or malfunctioning sensors; this work addresses specifically the funnel effect, a network problem caused naturally by the data flows themselves.

We first determine the candidate geographical placements for a relay node around the sink in terms of the locations of the sensors that need coverage. Each sensor and relay node has a transmission range r; therefore, for any node that falls within this range of another node may establish a connection. The relays cover sensor nodes from the sink to the nodes at the boundary of a congested region (the Convex Hull, or the ring) in order to alleviate the funnel effect. To guarantee the minimum number of relays used, the location are chosen according to the following ***placement condition***:

> *The geographical location of a new relay node is that which covers the largest number of elements of its convex hull and and it is the closest to the sink.*

Figure 1.6 shows the relay nodes R placed around the $Sink$ and covering all the sensor nodes that form a Convex Hull in the sensor network. In the context of this thesis, this scenario also

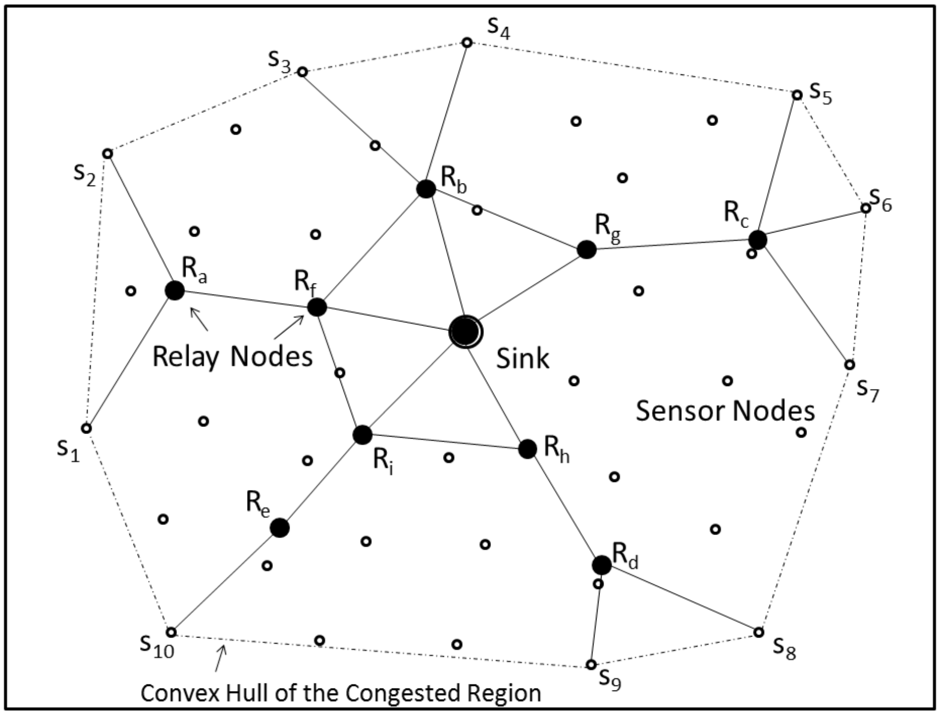


Figure 1.6: Topology example solved using The Placement Algorithm

shows potential for multiple paths that MPTCP can leverage in end-to-end connections from remote sensor nodes. Further details in [93].

1.6 Publications

This section provides a brief description of the publications supporting this thesis.

1.6.1 Mitigate Funnel Effect in Sensor Networks with Multi-Interface Relay Nodes

This prior work introduced in Section 1.5.2 represents the first step of this thesis into heterogeneous network scenarios. In a sensor network setting, nodes form a network and generate data flows directed to a single destination sink. When all the flows converge around the sink, high congestion occurs and the nodes' energy resources deplete at a fast rate, this is known as the *funnel effect*. The solution proposed is the addition of an wireless network overlay formed by mobile relay nodes capable of picking up data flows in an attempt to mitigate the funnel effect around the sink. For more details see [93].

1.6.2 A Cluster Based Architecture for Intersection Collision Avoidance Using Heterogeneous Networks

Vehicular networks in heterogeneous environments are explored in the work presented in Chapter 3. An urban vehicular safety application is proposed for vehicles organized into *platoons*, or a wireless ad-hoc network where vehicles are the nodes. While intra-platoon communication occurs over WiFi using ISM orthogonal channels whenever possible, communication among platoon leaders use LTE. This application allows platoons to exchange VANET CAM (Cooperative Awareness) messages among all vehicles without overloading the LTE network and without interfering over the ISM spectrum. CAM messages exchange vehicular status data such as the vehicle's type, position, velocity, acceleration, heading, etc., instrumental data for applications involved in vehicular safety on the roads. More details can be found in [117].

1.6.3 Multipath TCP in SDN-enabled LEO Satellite Networks

Low Earth Orbiting (LEO) satellite networks are part of the next generation 5G networks, a currently new research topic in after 4G LTE cellular technologies. Software Defined Networks, SDNs, help integrate the satellites with the terrestrial networks and in order to provide a smoother, more flexible network service and configuration. TCP flows, unfortunately, may suffer when operating in satellite networks due to long delays and frequent handovers. This work explores the use of MPTCP with the assistance of SDNs in end-to-end connections when frequent handovers occur in LEO satellite networks and helps find disjoint paths for MPTCP subflows to operate on. For further details see [33].

1.6.4 Multipath TCP on a VANET: A Performance Study

This study presents a test bed for a Vehicular Network, VANET, where MPTCP is analyzed, [92]. A mobile device, MPTCP capable, establishes an end-to-end connection with a remote, also MPTCP capable web server using two Internet paths via a WiFi network and cellular LTE. The vehicle operates around the city of Los Angeles, California, connecting to WiFi Roadside Units (WiFi RSU), or mobile WiFi access points, when the vehicle is in its range. The work studies the handover performance of MPTCP as it enters and leaves the WiFi RSU and LTE networks while the vehicle is under distinct velocities. Two perspectives are taken: Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V).

1.6.4.1 Multipath TCP on Vehicle-to-Infrastructure Scenarios

This work detailed in Chapter 5 narrows the scope of the study in Section 1.6.4 by taking a closer focus on Vehicle-to-Infrastructure, V2I, scenarios in a VANET. Under V2I, this work uses WiFi RSU access points to measure the performance of MPTCP as the vehicle enters and leaves its coverage. Moreover, this work also uses the Time Warner WiFi metropolitan network deployed throughout the city of Los Angeles, California. The vehicle's routes used in this work are longer and the MPTCP measurements are more comprehensive. Under distinct vehicle's velocity and routes, the study reports the performance of MPTCP and its viability as a transport protocol operating under high mobile scenarios.

1.6.4.2 Multipath TCP on Vehicle-to-Vehicle Scenarios

Chapter 6 presents the details of this work, similar to the work in Section 1.6.4.1 but with a focus on Vehicle-to-Vehicle VANET scenarios instead. The vehicles used in this work operate at higher velocities, including city and highway routes. However, unlike in the work in Section 1.6.4.2, the WiFi path used is not part of any infrastructure; instead, the vehicles establish a WiFi LAN between themselves to share their cellular LTE connections. Again, the MPTCP performance is observed and analyzed, and the work reports the viability of MPTCP under such scenarios and mobile characteristics.

1.6.5 MPTCP Path Selection using CapProbe

When MPTCP discovers a new path, for example, the mobile device associates with a new wireless network within proximity, it immediately connects to the end host and tries to allocate traffic load onto it. Mobile devices may be configured to use at most one path due to energy saving considerations, e.g., the Apple's iPhone smartphone. When a new WiFi network is discovered, it may be preferred over a cellular LTE network, but it does not necessarily mean that it is the best path. This work presented in Chapter 7 addresses the *path selection* problem in MPTCP capable mobile devices when they are constrained to use at most one interface.

1.6.6 Vehicular Network Operating on a Cognitive Network

This work presents two test beds: a wireless interference sensing test bed and an environmental pollution sensing system. The goal in this work is to understand the wireless interference caused by residential and commercial WiFi access points on VANET vehicles and the identification of WiFi access points for VANET vehicles. Vehicles operating in an urban environment make ideal candidates for a pollution sensing system that collect samples along the vehicle routes that are delivered to remote data centers for further analyzing. For more details see [94].

1.6.7 US Patent 9660710, 8868067: Identifying Coverage Holes Using Inter-RAT Handover Measurements

This set of patents correspond to the work performed during a 2012 internship at Intel, Corp., and they are detailed in Section 4.1. They discuss the concept of *coverage holes* in LTE networks, relevant to heterogeneous environments such as the VANET scenarios discussed in this thesis. In the LTE infrastructure, the Capacity and Coverage Optimization (CCO) functions are responsible to address the issues of cellular outage caused either by excess demand or sinister events. These set of patents address the detection of coverage holes and the enhancement of CCO functions under those scenarios. Further details in [25].

1.6.8 US Patent 9538413, 9154978, 8995255: Coverage Adjustment in E-UTRA Networks

Also part of the set of patents submitted as a member of Intel, Corp., in 2012, these patents address ways to detect a coverage hole using the measurement failure reports from the mobile cellular devices and the way to correct such an event. Described in Section 4.2, use case scenarios are large people congregation areas such as stadiums, shopping malls, or traffic jams. Cellular antennae may be readjusted to cooperate with spectral resources to the areas where a coverage hole has been detected. Further details in [24].

1.6.9 US Patent 9363727: System and Method of Cell Outage Compensation in Cellular Systems

This patent is a previous work related to US Patents 9538413, 8995255: Coverage Adjustment in E-UTRA Networks introduced in Section 1.6.8 and further detailed in Section 4.2. Further details in [23]

CHAPTER 2

Background

This chapter presents a small background of the VANET first; then it presents Multipath TCP and all its major concepts. Next we present some introductory material on the wireless technologies relevant to this thesis, WiFi and LTE, as well as important 3GPP specifications for ProSe and WLAN offloading, this latter relevant to MPTCP offloading. Finally this thesis introduces Cognitive Radio Network concepts relevant for wireless lossy link detection in Chapter 8.

2.1 Vehicular Networks

We say that we have a *vehicular ad-hoc network*, or VANET, when vehicles are equipped with on-board wireless communication devices that allow them to establish communication with other vehicles within their radio range [19]. The concept is very simple: "add a wireless interface to a vehicle and set up a wireless LAN (WLAN) that maintains a mobile network of moving vehicles" [53]. The VANET is an ad-hoc mesh network with highly dynamic nodes, where vehicles generate and receive messages either from other vehicles or foreign servers. Several radio technologies may be used to maintain connectivity, such as: WiFi [56], WAVE [57], DSRC [113], Cellular 3/4G LTE [4]. There are several applications that make VANETs desirable. The first and most obvious is to enhance the safety of passengers, the vehicles, and the road infrastructure. If vehicles are able to collect and exchange data from the road they operate in, other vehicles may take preventive measurements if dangerous situations lie ahead. Infotainment applications, or entertainment and information services provided to vehicles, is another example that includes services like video-on-demand (VoD) [85, 47, 126], cloud services [130, 12], or web and social networking.

A major research effort for VANETs is the Intelligent Transportation System, ITS [11]. ITS primarily focuses on the improvement of transportation safety and mobility, the reduction of the environmental impact, and the productivity enhancement of the transportation infrastructure. The USDOT describes five fundamental themes with significant impact to society:

1. enable safer vehicles and roadways;
2. enhance mobility;
3. limit the environmental impact;
4. promote innovation; and
5. support transportation connectivity.

In the ITS *Connected Vehicles* program, the VANET plays a major role. Two key communication infrastructures are expected to drive the research effort: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I).

Vehicle-to-Vehicle Communication (V2V) V2V refers to the direct connection between two vehicles, regardless of the roadside infrastructure. ITS bases this communication on DSRC [11]; alternative radio technologies can also be used. Several projects that involve V2V communications include [45, 5, 122], with a broad scope of applications.

Vehicle-to-Infrastructure Communication (V2I) A road-side infrastructure may be deployed in order to assist vehicles operating in both the urban or the rural environment. V2V communication may be complex and short-lived due to the vehicular dynamics so an accessible gateway that allows a vehicle to move traffic away from or across the VANET and into the Internet is convenient. Studies have analyzed the performance of a network that involves a V2I in [31, 32, 67]; some projects include [27].

According to [26], a Vehicular Network, or VANET, *refers to the possibility of having a communication node on-board a vehicle able to establish a wireless communication with other surrounding communication nodes within wireless range*. The basic idea is to use vehicles as mobile nodes to cooperate into exchanging data collected from their environment surroundings in order to generate knowledge that will collectively benefit the participants; see Figure 8.1.

The observation is that vehicles generally operate on mapped roads; therefore, they depict predictable trajectories, as opposed to pedestrians or, up to some extent, bicyclists. Moreover, vehicles are equipped with virtually unlimited resources during operation while still being mobile and fast; they are a sustained energy source that powers computational expensive devices (on board computers) capable of solving complex problems than mobile devices like the smart phones.

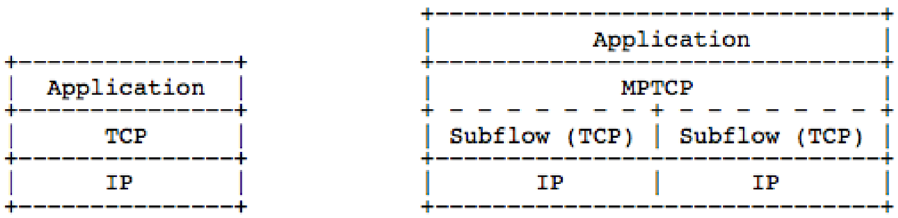
However, this notion of communicating vehicles has become outdated due to the enormous advances of radio communications and the proliferation of inexpensive commodity equipment. Radio technologies such as WiFi, cellular (UMTS/LTE), and personal networks (Bluetooth™) are widely available to mass population and are present in almost every smartphone currently in use. As opposed to old proprietary radio technologies and protocols from car manufacturers and stagnant standards such as DSRC (dedicated short-range communication systems) or, IEEE 802.11p, these newer radio technologies have significantly impacted the concept of what a VANET is, as described in [26]

> *The vehicle should no restrict its capability to communicate with other vehicles only, but should become able to connect with many different communication nodes that can be deployed or used in the road-side infrastructure remote centers, pedestrians, bikers, smart devices, etc.*

Here we describe some use cases for a VANET identified in [116]

1. *Intersection collision avoidance*. It refers all those applications related to a vehicle approaching an intersection; for example, warnings for signal violations, collisions, present pedestrians, etc.

2. *Public safety*. These applications target road assistants and drivers themselves in order to expedite help on the road; for example, approaching emergency vehicle warning, emergency vehicle signal preemption (turn traffic signals in favor of these vehicles), SOS services (incident is not collision), and post-crash warnings.



```
                          +-------------------------------+
                          |           Application         |
+---------------+         +-------------------------------+
|  Application  |         |             MPTCP             |
+---------------+         + - - - - - - - + - - - - - - - +
|      TCP      |         | Subflow (TCP) | Subflow (TCP) |
+---------------+         +-------------------------------+
|      IP       |         |       IP      |      IP       |
+---------------+         +-------------------------------+
```

Figure 2.1: MPTCP architecture is based on independent TCP Subflows.

3. *Sign extension*. These use cases refer to the information augmentation of the road signs. For example, a sign might be obstructed by the natural elements, such like a tree or debris. A smart city may notify the incoming vehicles of the existence of important signals that the driver may have omitted to realize. Alerts for wrong-way drivers, working zones, or low bridge coming up may be a few examples.

4. *Vehicle diagnostics and maintenance*. A safety recall that has been just announced takes some time to reach the vehicle owners; this is an example of how a vehicle manufacturing company may reach all their customers to warn them of issues with their vehicles. Other examples are just-in-time repair notifications, with quote prices and estimated repair time.

5. *Information from other vehicles*. This use case is one of the most beneficial since it encourages vehicles to share recently learned knowledge of the road and the environment with other VANET participants. Examples include road-condition warnings, high-way, rail collision notification, highway lane merge assistant, cooperative adaptive cruise control, etc.

2.2 Multipath TCP

Multipath TCP [44], MPTCP, is a direct evolution from TCP that allows an end-to-end session to establish a connection between two hosts using one or more interfaces concurrently. Contrary to TCP, MPTCP can maintain its connected state in the event of a wireless handover, link failure, or during any increase or decrease of available interfaces on the host. Moreover, because it is based on TCP, it is backwards compatible with the current network establishment, as opposed to competing technologies such as SCTP [105], Snoop TCP [9], or Wireless TCP [108].

We now continue with a short description of the Multipath TCP protocol and the Resource Pooling principle; we then identify the important components in MPTCP and provide a short explanation of each.

2.2.1 Resource Pooling

The *resource pooling* principle [124] refers to the idea of managing the network resources as if they make up a single pooled resource with the following benefits:

1. improve in network robustness;
2. ability to perform load balancing through traffic engineering;
3. maximize network utilization.

In packet networks, this principle is partially achieved when flows use the entire link capacity between nodes at a transmission, as opposed to circuit-switched networks where each flow receives a fixed, partial capacity of the link. It falls short, however, when TCP fails to take advantage of other paths available in the network when a transmission burst on the ongoing TCP session demands larger resources than its current path's capacity is able to offer.

MPTCP implements the resource pooling principle by maintaining a subset of pseudo-independent TCP subflows at the transport layer, all controlled by a coupling protocol that greedily schedules packets on its current subflows and discovers new or terminates failing paths in the network. When two end hosts are both MPTCP capable, that is, both implement MPTCP, and at least one has multiple network connectivity (Ethernet, WiFi, Cellular, etc.), then MPTCP is able to leverage the end hosts paths to improve end-to-end communication. However, if one end host does not implement MPTCP, then the communication falls back to traditional TCP, thus making MPTCP backwards compatible. Depending on the congestion control protocol used, the TCP subflows may be completely autonomous but unfair to other TCP flows (UNCOUPLED) or dependent on each other's current performance (COUPLED, LIA, OLIA) [123, 77]. By maintaining a set of TCP subflows, MPTCP coherently couples the resources its multiple paths provide as if they all would constitute a single fat network pipe.

2.2.2 Path Manager

The *Path Manager* is the functional component that detects and signals multiple paths between hosts, [43]. It identifies a path by the standard TCP five-tuple: (source address, destination address, source port, destination port, protocol); thus, allowing the protocol to differentiate paths by both IP addresses and port numbers. Once a new path is added to a particular interface, it tries to establish a new TCP subflow and begins to schedule packets on it.

On the current MPTCP implementation in Linux, [96], available paths are detected via the following network interface events: NETDEV_UP, NETDEV_DOWN, and NETDEV_CHANGE. It is important to point out that if the Path Manager catches NETDEV_UP event, MPTCP blindly schedules packets without prior knowledge of the path. The reason for this behavior is probing; i.e., if the path is lossy or unstable, MPTCP probes the path hoping that it will eventually improve so it can later transfer some load from other more congested paths. Current path managers implemented in Linux may be found in [96].

default. This is a passive path manager; it does not announce nor it tries to establish multiple subflows with its connected peer. However, should the other end initiates another subflow, it allows the multipath connection.

fullmesh. This path manager attempts to establish a subflow for every combination of pairs of addresses between the source and the destination. Moreover, it may be configured to establish more than one subflow per path; current setting is one subflow per path.

ndiffports. This path manager uses a single IP-address pair and multiplexes the network connection via the port numbers. The number of ports, and thus the number of paths, used is a configurable parameter.

binder. This path manager is an architecture that uses Loose Source and Record Routing to capture packets from users connected via an access point. Packets are then relayed using MPTCP and tunneling, making up an community network that aggregates several Internet Gateways, [14].

2.2.3 Scheduler

The *Scheduler* is a fundamental component of Multipath TCP. It takes a byte stream from the higher layer's buffer just as TCP would do and breaks it up into segments that are allocated onto the available TCP subflows [43]. Unlike TCP, MPTCP has multiple paths onto which segments may be delivered; therefore, a mechanism that maintains the logical ordering of the segments when they arrive at the other end on distinct paths is necessary. MPTCP uses its own data sequence mapping mechanism encoded on the TCP header options field [44] for this purpose. In [129], the authors identify two decisions that an MPTCP scheduler must take in order to send a packet: (1) which subflow to use; and (2) how much data data should be sent on the subflow. Current schedules available are described in [96].

default. The default scheduler allocates packets on a subflow that currently has the lowest recorded smooth RTT value, until its congestion window is full. Like a waterfall, it will continue with the next best subflow, so on and so forth.

roundrobin. This scheduler tries to allocate packets on all the available paths in a round-robin fashion. It is configurable in the number of packets delivered per turn and if it should fill the congestion window or not in order to truly achieve a round-robin scheduling.

redundant. This scheduler sends a packet on all interfaces in order to minimize latency at the expense of throughput.

2.2.4 Congestion Control

The congestion control algorithm plays a major role in the MPTCP performance. Just like TCP [69, 87], MPTCP subflows maintain a congestion window, perform flow control and keep retransmission timers; they also perform the three-way handshake during connection establishment and the slow start algorithm at the beginning of a transmission or a timeout. They differ, however, on the congestion avoidance algorithm used. An MPTCP congestion control algorithm strives to meet the following goals [17]:

1. *Improve throughput*: This is a lower bound that states that the multipath algorithm should

perform at least as well as a single-path TCP operating over the same bottleneck.

2. *Do not harm*: This is the fairness requirement and it states that a multipath flow should not take more network resources than any single-path TCP flow on any path.

3. *Balance congestion*: Given the multipath nature of MPTCP, traffic load should be taken off congested paths into less congested ones, subject to meeting the first two goals.

A closed-form equation for the first two goals is derived in [123] as follows:

$$\sum_{r \in R} \frac{w_r}{\mathrm{RTT}_r} \geq \max_{r \in R} \frac{\hat{w}_r^{\mathrm{TCP}}}{\mathrm{RTT}_r} \tag{2.1}$$

$$\sum_{r \in S} \frac{w_r}{\mathrm{RTT}_r} \leq \max_{r \in S} \frac{\hat{w}_r^{\mathrm{TCP}}}{\mathrm{RTT}_r}, \quad \text{forall } \; S \subseteq R. \tag{2.2}$$

A subflow $r \in R$ is part of a set of subflows R operating in an MPTCP node, where each subflow maintain a congestion window w_r and experience a round-trip time RTT_r; additionally, $\hat{w}_r$ and $\hat{w}_r^{\mathrm{TCP}}$ are the equilibrium windows obtained when a subflow and a single-path TCP flow respectively experience the path r's loss rate. Also, $S \subseteq R$ is any subset that contains the path r. The fairness equation 2.2 does not imply that a Multipath TCP connection should never exceed a single-path TCP performance; it means that in the absence of competing flows, MPTCP should take all the underutilized network resources. When competing traffic is present, it should fairly share resources with other single-path TCP flows experiencing the same loss rate. For more details, see [123].

We now summarize the available congestion control algorithms in [96]; the table shows the behavior of the congestion window upon a ACK arrival or loss detection. All the algorithms perform a slow-start stage at the beginning and during a retransmission timeout events, just as TCP would do; the difference lies mostly during the congestion avoidance stage.

Name	ACK	Loss
TCP	$1/w$	$w/2$
COUPLED	$1/w_{total}$	$w_{total}/2$
UNCOUPLED	$1/w_r$	$w_r/2$
LIA	$\min(\frac{\max_{s \in S} w_s/\text{RTT}_s^2}{(\sum_{s \in S} w_s/\text{RTT}_s)^2}, \frac{1}{w_r})$	$w_r/2$
OLIA	$\frac{w_i/\text{RTT}_i^2}{(\sum_{r \in R} w_r/\text{RTT}_r)^2} + \frac{\alpha_i}{w_i}$	$w_r/2$

Table 2.1: MPTCP Congestion Control Algorithms

2.3 IEEE 802.11, WiFi

Undoubtedly, WiFi is the most popular wireless LAN with the largest market penetration. It consists of a half-duplex network operating on the 2.4GHz and 5GHz ISM unlicensed band of the radio spectrum. Implementations of the physical layer consist on Frequency Hopping Spread Spectrum (FHSS) and Direct-Sequence Spread Spectrum (DSSS) in the original standardization 802.11 [100]. Later enhancements include the use of Orthogonal Frequency Division Multiplexing (OFDM) in 802.11a operating at 5GHz, High-Rate DSSS in 802.11b and Extended-Rate Physical in 802.11g. The latest physical layer implementation is 802.11n that includes MIMO technologies with OFDM. At the MAC layer, the 802.11 MAC shares similar concepts as Ethernet, particularly the addressing scheme:

> *Wireless network interface cards are assigned 48-bit MAC addresses, and, for all practical purposes, they look like Ethernet network interface cards. In fact, the MAC address assignment is done from the same address pool so that 802.11 cards have unique addresses even when deployed into a network with wired Ethernet stations* [46].

The WiFi architecture is the following. *Stations* represent the mobile devices connected wirelessly with the goal of exchanging data messages. They may be any device with any computational characteristics equipped with a wireless interface card. Even though stations are mostly mobile, there is no reason why they must be mobile. The only characteristic that makes up a station in terms of the WiFi architecture is that they connect with its peer via the wireless interface card.

The *Access Point* is the gateway between the wireless WiFi infrastructure and the outside world. When a station needs to send a message to a node outside the WiFi network, there must be a translation of the 802.11 message into the protocol that the outside world uses. The access point performs this function. Also, access points are responsible of the coordination of the stations in the network. For example, the WiFi network might be configured as a contention-based accessing scheme or a token-based scheme and the access point is responsible the proper operation of these types of networks.

The *Wireless Channel* is the medium that the WiFi network uses to connect over the air. There are 11 channels used in the 2.4GHz ISM band, three of which are orthogonal (non-overlapping) and many more 20MHz slots that span the 5GHz band. This last band is much more regulated and its availability may be affected by the local policies of each country where the network operates on.

Finally there are two types of WiFi networks: *infrastructure* mode and *ad-hoc* mode. The first relies on the access point to coordinate the operation of the network. Any communication among stations must use the access point to deliver its messages. In an ad-hoc mode, there is no access point. Communication among the peer is relayed among the nodes of the network until the destination node is reached.

2.4 Cellular Networks

Radio Access Technologies (RATs) or Cellular Networks are wireless networks operating in a strictly infrastructure mode where the Mobile Station, or the cellular device, connects to the core network (the wired network) via an Base Station, or the antenna or tower, that the carrier company installs in the metropolitan area. The towers use usually three directional antennas that cover the area surrounding it forming a cell of coverage where the mobile stations may roam freely; hence the cellular term. If a user desires to move outside one cell into another without loosing its connectivity, it must perform a “handover” of the connection. The core network may performs this procedure when the mobile station leaves the area of coverage or the base stations may perform this procedure locally. In an abstract view, a group of base stations span the metropolitan area of coverage fully covering it with cells in order to allow the mobile station stay connected as it moves freely across the city.

2.4.1 LTE

The Long Term Evolution, or LTE, cellular network is a 4G technology consisting of an all-IP access technology (packet-based only) based on OFDMA as its underlying RAT. With the information technology revolution of the last decade the mobile broadband market has increased its demand for services. For this reason, LTE has been redesigned from the physical layer in order to meet the IMT-Advanced requirements [28]. LTE can be considered as a complementing expanding technology to the GERAN and UTRA voice and data services; it includes a new system architecture called System Architecture Evolution (or SAE) that in conjunction with the redesigned radio interface makes a IP packet-switched network suitable to naturally bond with the current IP network and Internet. The following is a list of requirements for the LTE [65]:

1. reduced delay, connection establishment and latency;
2. increased user data rates;
3. increased bit rates at the edge of the cell;
4. improvement on the spectral efficiency;
5. greater flexibility on the spectrum usage;
6. simplified network architecture;
7. seamless mobility across distinct RATs;
8. low power usage.

2.5 Standardization and Specifications

3GPP is a consortium of companies developing specifications for next generation Wide Area Networks WAN, specifically cellular LTE. It has published two particular specifications related to heterogeneous networks. The first is *LTE ProSe* [2]; it describes the use cases of how two UE devices should find each other when they are within proximity. The second is *LTE WLAN Offloading*

[3], which also provides the procedures and use cases on how a UE device should switch from LTE networks to WLAN, and vice-versa.

2.5.1 3GPP TR 22.803, Feasibility Study for Proximity Services (ProSe)

Proximity Services [2] in 3GPP networks refer to the identification, discovery, and communication establishment of User Equipment (UE) devices using the network operators based on their proximity. *Proximity* of UE devices is defined by the *ProSe Discovery* function, which in turn is defined by the network operator's criteria. For example, a network operator may define that two UEs are in proximity with each other if they are within 500m distance; therefore, if these two UEs are *ProSe-enabled*, they may activate the *ProSe Discovery* function and establish a *ProSe Communication* link. The LTE ProSe specification is a study that identifies the use cases and provides the service requirements for an network-operated, controlled discovery and communication between UEs that are within proximity. The main applications for *proximity services* are listed next.

1. Commercial and social use
2. Network Offloading
3. Public Safety
4. As a support service for other services in the infrastructure

This proximity discovery would allow the LTE network to provide a more efficient service in terms of resource optimization. For example, if two UE devices are within proximity, the LTE network, maintaining total control of the UE at all times, may signal the UEs to activate LTE WLAN Offloading and carry on with their communication using WiFi while releasing LTE resources, see Figure 2.3 and compare it with Figure 2.2.

Moreover other UEs configured with LTE ProSe may find each other automatically and exchange information without the user to intervene; automated discovery and organization is a fundamental feature of many mobile systems such as VANETs. In Public Safety, some of the services

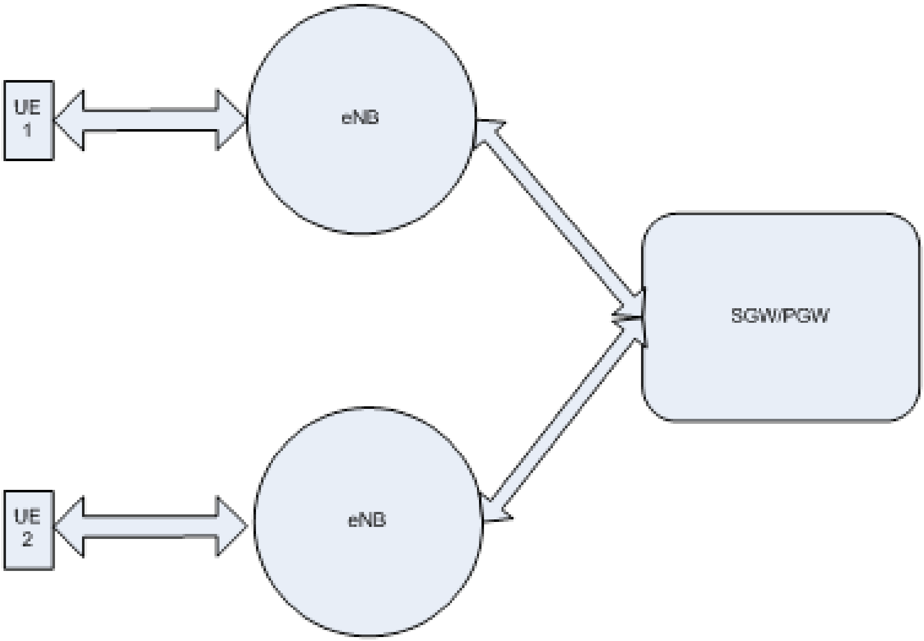


Figure 2.2: LTE UE communication: current architecture

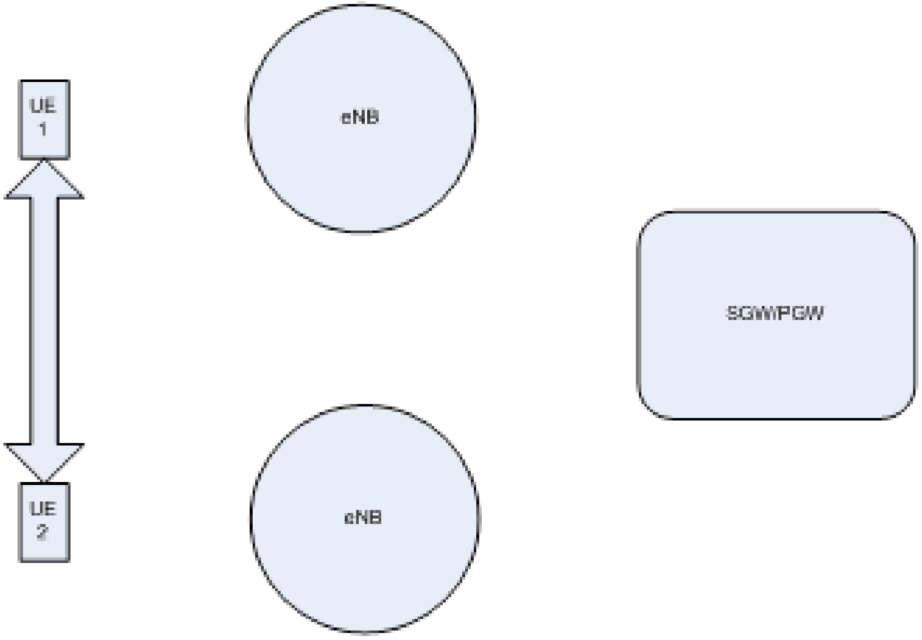


Figure 2.3: LTE UE propose architecture: UE to UE communication

that ProSe may facilitate are the following: build and organize groups, provide coverage range extensions, perform collective service discoveries, relay data gathered when LTE coverage outages.

Mobile device discovery may be used to find WLAN hotspots if they are controlled by the network provided. This WLAN discovery has an impact on the service continuity and multipath selection problem of this thesis.

2.5.2 3GPP TR 23.890, WLAN Offloading

WLAN Offloading, [3], provides use cases and solutions to scenarios where LTE offloading to a WLAN could take place. We identified key use cases relevant to this thesis:

1. ping-pong offloading to WLAN (oscillating behavior);
2. WLAN preferences with respect to specific RATs;
3. undesirable LTE connection loss, degradation or suspension due to a handover.

From all the studies and analysis that the technical report provides, this thesis focuses on the following area: mobility and radio access technology handovers between a 3GPP LTE, UTRAN (4G), or WLAN. It is often the case that an LTE connection suffers due to high demand in a cell. Examples of this scenario may be the shopping center. However, it is also often the case that some of these scenarios have WLAN base stations that provide network access to the Internet; for example, restaurants and coffee shops at malls or airports. A UE device may take the decision to switch its data flow from LTE to WLAN or stay in LTE upon detecting an available WiFi access point. This is unfortunately not an automated service. If the user decides to switch the connection, a hard handover is usually performed; that is, the user needs to disconnect from one network and reconnect onto the other. In this scenario, currently running connections and services suffer (and may disconnect). Moreover, if the user decides to perform the handover, it is possible that the UE may connect with the access point but discover that this access point also suffers from large congestion or interference. In this situation, the handover failed because it moved the connection to a under-performing network. At the time the user realizes this, she may decide to switch back to the cellular LTE connection and once back in the LTE network, another handover may be suggested to another untested WiFi network. Some smartphones may include programs that perform the hand over automatically. However, they just change automatically to the new discovered network as soon as the connection is established, without performing a connection test and compare it with the current one. Typically these decisions are based on predefined user preference rather than network performance.

These are some of the issues that WLAN offloading faces that are addressed on this thesis. The benefits of offloading the LTE network is to, first, reduce congestion in a wide area network, and second, save on the costs since cellular networks are a paid service. Handovers from LTE to WLAN, operated and controlled by the network, are related to the service continuity and multipath selection.

2.6 Cognitive Radio Networks

A cognitive radio network, or CogNet, is a network of computers equipped with radio transceivers capable of adapting its connectivity based on the knowledge learned about the environmental where it operates, [37]. The concept of Cognitive Radio is first coined by Joe Mitola III in his PhD Thesis, [58],

> *The term cognitive radio identifies the point at which wireless personal digital assistants (PDAs) and the related networks are sufficiently computationally intelligent about radio resources and related computer-to-computer communications to (a) detect user communications needs as a function of use context, and (b) to provide radio resources and wireless services most appropriate to those needs.*

The research community quickly realized the potential of cognitive radio networks to such a point that regulating bodies, such the International Telecommunication Union, ITU, formally defined the concept in [64] as follows:

> *A radio system employing technology that allows the system to obtain knowledge of its operational and geographical environment, established policies and its internal state; to dynamically and autonomously adjust its operational parameters and protocols ac-*

> *cording to its obtained knowledge in order to achieve predefined objectives; and to learn from the results obtained.*

For this project we decided to use a CogNet in order to understand better how to improve the communication among the participants of a VANET. It may be possible that vehicles, pedestrians, bicyclists, motorcyclists, etc., may participate actively on a VANET; however, different issues affect the communications of these subjects when they transmit data over the wireless medium. One of these issues may be the interference that a transmitter may experience from sources around it. For example, when a vehicle drives close to a residential or a commercial zone, it is possible to detect several wireless hot spots on the street. Moreover, during daytime the traffic on cellular networks is characteristically high, which may negatively affect the VANET and may limit communication to rely on cellular networks; thus, creating an unnecessary dependency.

One of the goals of this project is *to study the interference that a participant of a VANET would experience on the road*. We built a test bed, see Subsection 8.3.1, that measures the amount of wireless interference onto the spectrum that our wireless transceivers use in order to understand how this issue affects our communications. A better understanding of the wireless interference will lead to better algorithms that move our communications around the wireless medium and thus fulfill truthfully the concept of a Cognitive Radio Network, CogNet, operating on top of a VANET.

CHAPTER 3

Heterogeneous Networks

Mobile devices are able to communicate and organize into networks using using radio network technologies such as WiFi, Bluetooth, and cellular LTE. Each radio technology possess distinct characteristics that make them unique; for example, WiFi is a well-known standard technology that allows free organization on a publicly available frequency band for a distance range below the hundreds of meters. On the other, LTE is a privately deployed network from the 3GPP consortium company members that use licensed wireless spectrum for a profit; as such, users pay for its use wherever there is coverage; it is common nowadays to find very good LTE coverage at major urban locations. A heterogeneous network leverages distinct radio access technologies to benefit their users. This chapter we show how a heterogeneous network helps solving a common safety VANET problem: collision avoidance of vehicles at traffic intersections.

3.1 Introduction

Passenger and pedestrian safety are important goals on the algorithms and protocols we design for Vehicular Networks. In urban areas, a common problem in VANETs is car collisions at street intersections. There are several ways to address this issue, but first we need realize that the problem of vehicular communication and coordination is a delay-sensitive one, that is, the vehicles must act at the shortest time possible; it is also a communications problem.

To address this issue, we can design a protocol that allows the cars to communicate among themselves in order to pass on information regarding the moving vehicle's characteristics, understand the overall intersection's scenario, and assess a collective solution in an expedite manner. To achieve such a goal, we typically assume that cars are equipped with wireless communication technology, be it broadband or local, that cars can use to exchange the protocol messages. The characteristics of an urban scenario with moving vehicles pose specific challenges to the radio technologies that must be considered. For example, vehicles could move at high speeds or in a stop-and-go traffic; vehicles also may suddenly become out of range from a radio transmitter and be reachable again due to large objects such as buildings.

It is common that vehicles in VANETs are equipped with commodity WiFi interfaces; it is also possible that they may carry a metered broadband interface, such as cellular or WiMax, that users pay to use. The advantage of using a WiFi network is that it is easy to use, convenient and free of cost. However, it suffers from traffic overload, signal degradation due to the large number of obstacles that the urban scenario may present, or its small coverage range for vehicles. On the other hand, nowadays we have LTE cellular technology, a broadband technology able to reach vehicles and service them access to the Internet regardless of the velocities they operate on. LTE suffers, however, with large traffic demand, especially during normal rush hours, and it incurs into a cost per usage. We concluded that a single radio technology operating in a VANET alone in the urban scenario is not ideal; in fact, they are complementary. A solution for vehicle coordination and communication for collision avoidance is one that leverages of the heterogeneous radio technologies available at the vehicles' disposal.

We address the problem of vehicle collision prevention and avoidance at an intersection by establishing a communication protocol using both cellular LTE and WiFi. The goal is to timely warn events that currently occur at any given intersection in the urban scenario to approaching vehicles. Instead of broadcasting messages as vehicles approach to an intersection, vehicles form clusters and select a cluster head to perform the delivery of the messages among themselves. For local message dissemination, the vehicles use WiFi, since it is free and it is short range communication. For long distance traffic transmission, the cluster head would be the only node allowed to use its LTE interface to reach the other vehicle clusters in the intersection. For example, vehicles collect local information among themselves using their WiFi interfaces. Next, they aggregate at their head node that uses its LTE interface to exchange these packets with other head nodes. Finally each head node disseminates the data back to its peer vehicles. This way, the LTE use is minimized only for inter-cluster communication while the data dissemination and collection is done using WiFi.

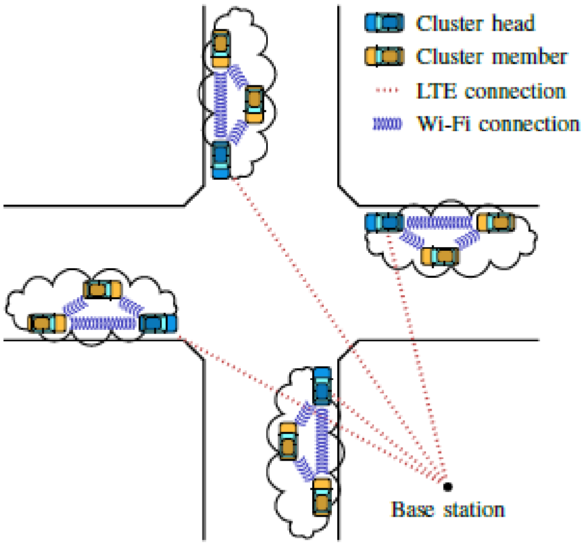


Figure 3.1: Node clusters exchange messages using WiFi within their cluster and use a cluster head for broadband dissemination.

3.2 Main Idea

The main idea is to organize the vehicles approaching to the intersection into clusters called "platoons". As Figure 3.1 shows, vehicles organize into clusters using WiFi for internal communication. A cluster head is selected depending on how close a vehicle is from the intersection; membership to a cluster depends, in turn, on the cluster head's range, which the vehicles know via the beacons. This way, if another vehicle receives a cluster heads beacon but it does not fall into its range, then it may form another platoon that follows the previous one. Since the cluster is always transmitting its beacons, it maintains its status as cluster head, but if it disappears from the platoon (perhaps the vehicle made a turn), other vehicle's will time out and declare themselves as cluster head and the process starts again.

One important aspect with WiFi is that clusters use distinct, non-overlapping, channels for inter-cluster communication whenever possible, see Figure 3.2. The reason is to avoid inter-platoon interference as much as possible. Given a frequency band such as 2.4GHz or the 5 GHz, a number of WiFi channels are defined. For example, in the ISM band there are eleven WiFi channels available for use, out of which only three are non-overlapping. We want to minimize the overlapping as much as possible, so in a standard intersection of four corners we allocate four channels with

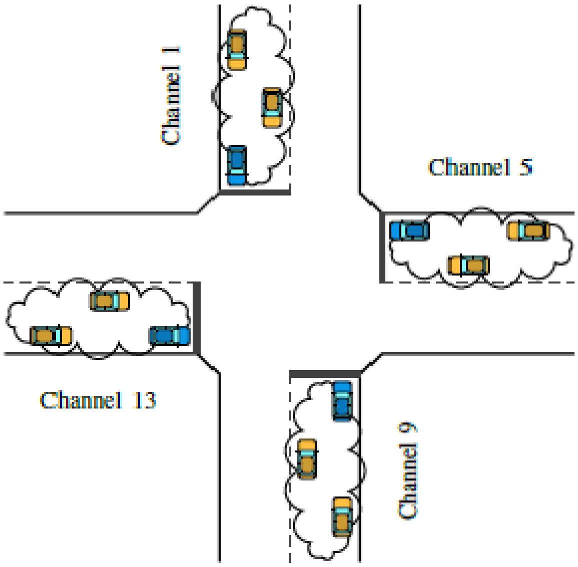


Figure 3.2: Cluster heads may use LTE to determine which WiFi channels they use before they arrive to the intersection in order to avoid wireless interference.

one platoon on each road. In general, the channel gap is defined as the discrete channel spacing between allocated channels to a platoon. That is simply the division of the number of available channels by the number of roads converging onto the intersection. For example, if there are 13 available channels, the channel gap between two allocated channels to platoons is $13/(4-1) \approx 4$, for a 4-leg intersection.

3.3 Results

Our findings are as follows. In Figure 3.3 we learned that as the number of vehicles increase, the use of a heterogeneous network (WiFi + LTE) outperform the single use of either WiFi or LTE. This is expected since the bulk of traffic from sensing vehicles is carried out by a private WiFi network operating on an orthogonal channel; thus, operating virtually free of interference considering that neighboring access point's interference negligible. Also, only platoon aggregated data is reported, not each sensing vehicle's. Moreover, this figure does not take into consideration of the cost per byte transmitted by each vehicle. Any decrease in the LTE data transmission would be translated in a decrease of unnecessary transmission costs; our algorithms try to minimize this

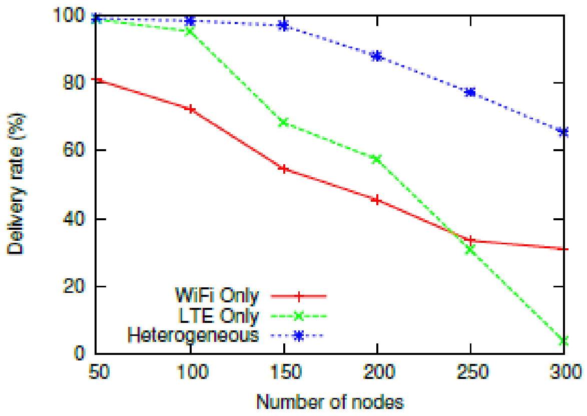


Figure 3.3: Delivery rate of CAM messages among vehicles

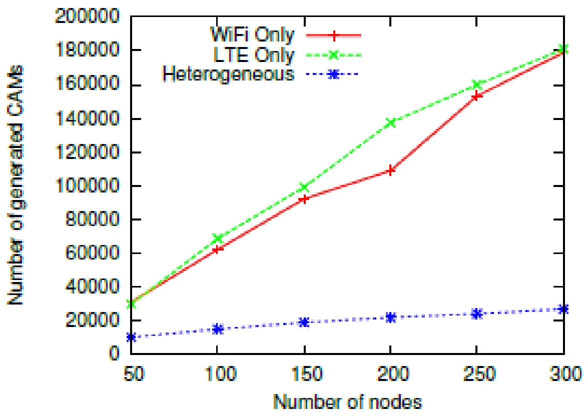


Figure 3.4: Traffic overhead at the cluster heads

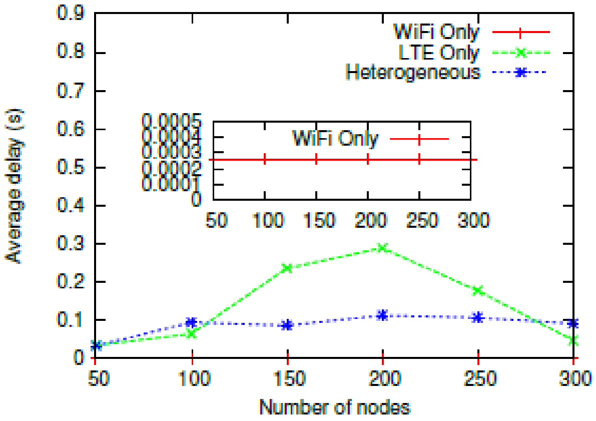


Figure 3.5: Average delay per radio technology

cost by leveraging on WiFi for inter-vehicle communications.

Figure 3.4 shows the traffic overhead in the delivery of CAM messages, or messages that deliver vehicle geographical information and motion dynamics to the entire network. Again, in a heterogeneous environment, these packets are aggregated and delivered only by the cluster head. Because the cluster head performs aggregation, some of these CAM messages are suppressed since they are not relevant to the rest of the network outside the platoon. This is what accounts for the decrease of data transmission by the cluster heads.

Figure 3.5 shows the average delay experienced by the heterogeneous network, the WiFi only and the LTE only settings. Unsurprisingly it shows that WiFi only scenarios experience the least delay, since the vehicles are always next to each other, while the use of LTE involves the message to travel from a node to the cluster node, then to another cluster node and to the destination vehicle in the other platoon. This means that the message goes up to the LTE network and back down again to the platoons; passing though network routers and competing with other LTE traffic. Therefore it is evident that the LTE delay dominates in both the heterogeneous network and the LTE only setting.

3.4 Relevance to the Thesis

In this chapter we addresses a VANET problem using a wireless heterogeneous network with a service that needs continuity and has delay requirements. We identify an application aiming to provide vehicular traffic safety with some delivery requirements. Our solution proposed leverages on the selection of networks with the best path characteristics; in our experiments, the networks used are LTE and WiFi, two popular and well established technologies in the market. This work also provides a study on how a protocol should be designed, based on the characteristics of wireless interfaces. We demonstrated that heterogeneous networks must be taken into consideration in the decision of how data is delivered on mobile devices with multiple interfaces. This work is an example of a application service that is not concerned with what happens below the application layer, more specifically, in the network, link, and physical layers. Our system autonomously explores the use of multiple interfaces in vehicles to provide continuity to the application it serves, which is a fundamental goal of this thesis.

CHAPTER 4

3GPP Patents

The work presented in this chapter is the result of the author's contribution to an internship at Intel, Corporation. The author worked with a dedicated and capable team eager to enhance the current 3GPP specifications for the Evolved Universal Terrestrial Radio Access (E-UTRA) wide area network, the foundation technology for LTE. The result is a set of patents coauthored with Joey Chou, from Intel Corporation, which are presented next.

4.1 Identifying Coverage Holes Using Inter-RAT Handover Measurements [25]

In cellular networks, a cell refers to the area where a mobile device receives service. In LTE terminology, a base station that serves in a cell is called an enhanced NodeB or eNB and it provides coverage to mobile devices called User Equipment, or UE. Urban scenarios where multiple obstacles are present, i.e., buildings, towers, commercial signs, etc., may block the signal of a eNB and create an empty coverage spot; if an UE does not receive a strong enough signal, it disconnects. This area is known as a *coverage hole*.

When an UE moves around the urban environment, the cellular system handles the mobile device's handover from cell tower to cell tower. When the Radio Access Technology (RAT) currently controlling the UE, say LTE, fails, it needs to be substituted by another RAT connection, say 4G, EDGE, or GSM, in order to continue connectivity; in such scenario, an inter-RAT handover occurs. During this inter-RAT procedure, the UEs collect measurements and generate reports that are continuously delivered to the system and forwarded to a Network Management entity, or NM, for further processing. This NM entity, provides the system with management functions that include

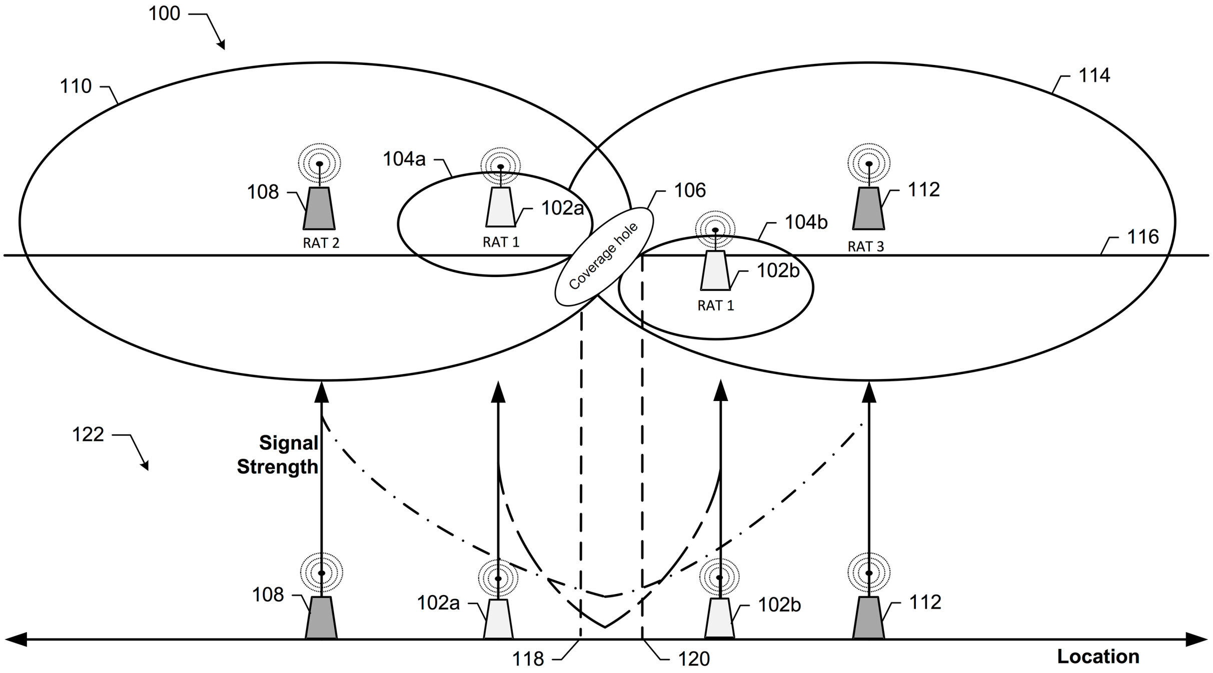


Figure 4.1: Coverage Hole use case during a inter-RAT handover.

coverage analysis and correction-action procedures, collectively known as Capacity and Coverage Optimization functions, or CCO. A coverage hole report issued by an UE in response to a cell tower handover may include the coverage hole's geographical location and its size, the reference signal received power (RSRP), a reference signal received quality (RSRQ), a cell tower identifier, among other information. The NM can analyze and correlate the reports to a specific geographical area and temporal location and conclude that it has identified a coverage hole; and issue a corrective action.

Figure 4.1 shows the use case for a moving UE through two cell towers depicted in the figure as 102a and 102b, from the same RAT1, for example, LTE. The UE travels along the path depicted as 116 from either direction. When the UE enters in the coverage hole, a handover onto RAT2 (4G, EDGE or GSM) happens and the UE generates a measurement report that the NM can use. Note how the signal to the base station in RAT2 and RAT1 in the perspective of the UE decreases as it approaches the coverage hole; this information is reported to the system for further analysis.

Figure 4.2 shows the architecture that the LTE uses to provide service to the mobile devices. UEs generate intra-RAT reports at the mobile UEs. These reports are forwarded to the eNBs, or

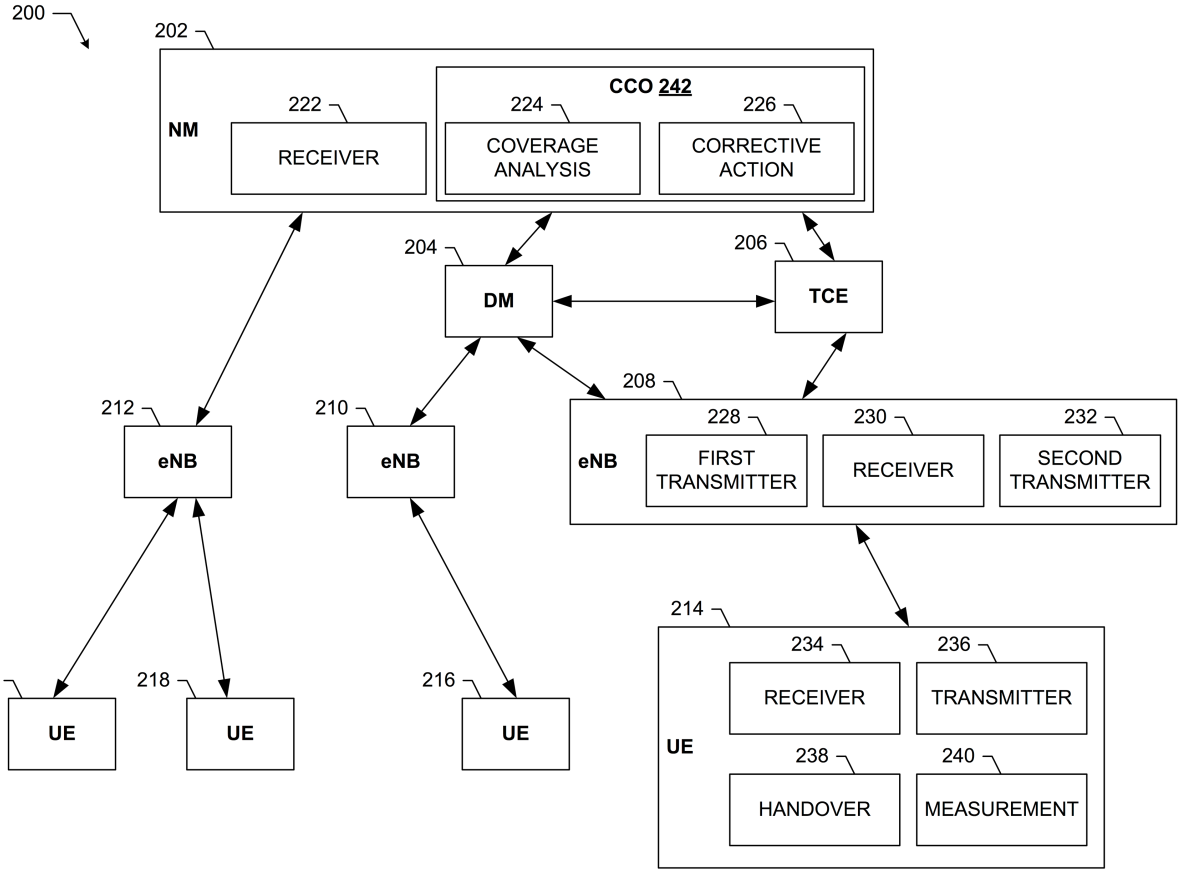


Figure 4.2: The E-UTRA service that LTE uses through which inter-RAT reports generated by UEs travel to reach the NM.

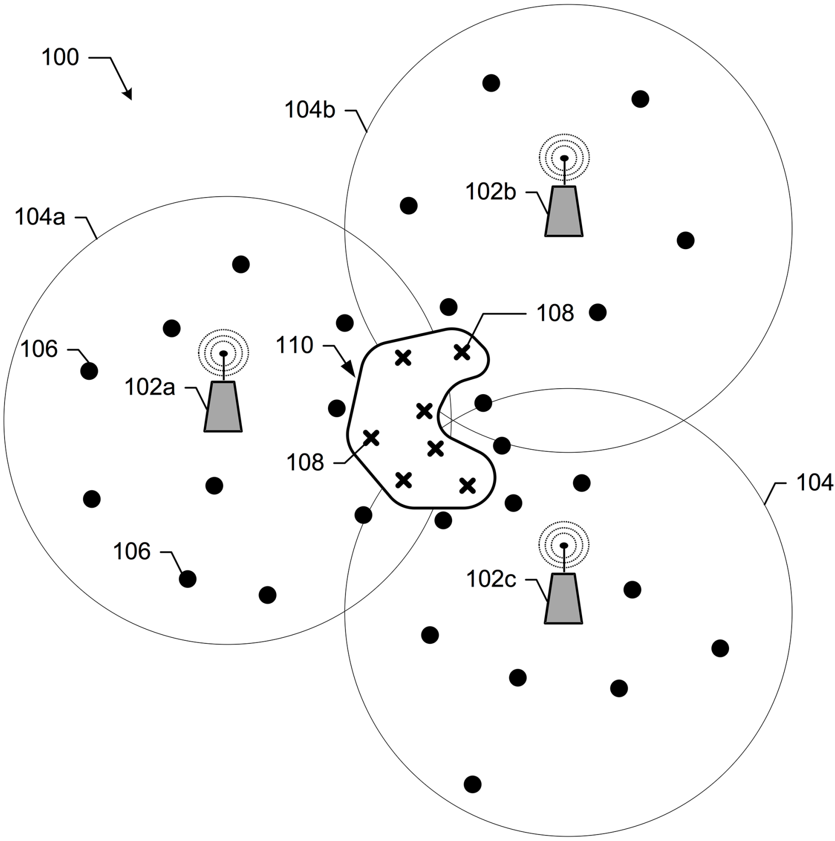


Figure 4.3: A coverage hole is identified with inter-RAT reports of bad signal quality shown here with an ×.

the cell towers, which may be directly connected to the NM or be handled by a Domain Manager (DM) labeled as 204 in the figure. In other scenarios, a Trace Collection Entity (TCE), labeled as 206, may collaborate with a group of eNBs and DM to collect such reports and forward them to the NM for further processing. The NM collects, analyzes, and issues a corrective action based upon its findings.

4.2 Coverage Adjustment in E-UTRA Networks [24, 23]

A cellular network is composed of coverage cells that provide service to UEs inside their coverage. As a paid service, users expect immediate connectivity once they open their devices; however, it may not always be the case. Connectivity service in the network may be compromised when a coverage hole appears; situations that could make a coverage hole are weak signals due to obsta-

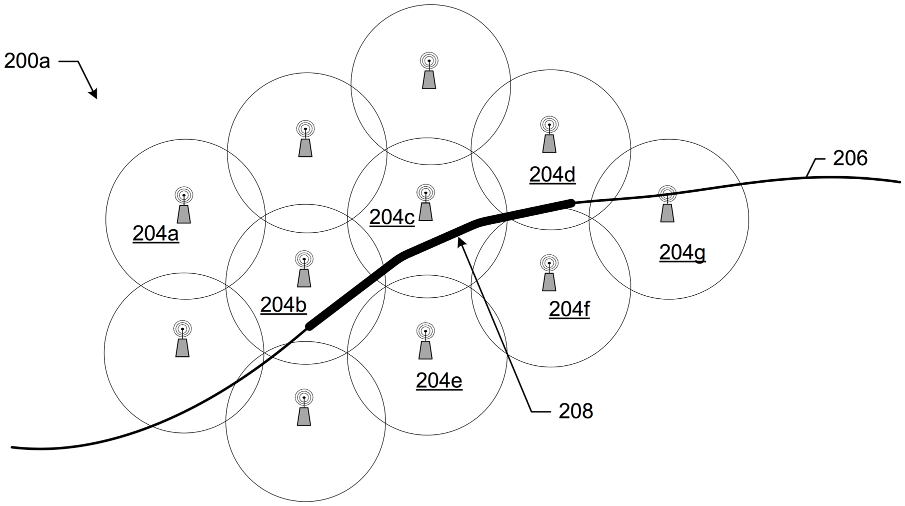


Figure 4.4

cles such as buildings or geographical areas at the cell's edge. Another situation that may lead to poor user experience are network overloads during rush hours or areas with large people congregation, i.e., sport stadiums, shopping malls, or traffic jams. In these scenarios, local cellular towers may become overloaded due to large traffic demand or coverage holes; however, sometimes these situations are localized, that means, neighboring cellular towers using network resources could help.

Figure 4.3 shows how a coverage hole is identified. Inter-RAT reports issued by UEs are generated at locations where bad signal quality forces the UE to handover to another RAT; that is, from LTE to 4G or even GSM. The dots in the same figure are reports with satisfactory service. Once these reports are forwarded to the NM, the coverage hole may be identified.

Figure 4.4 depicts the case when a cell may become overloaded. In environment 200a, a traffic jam 208 in a high way scenario, freeway 206 is serviced by cellular towers 204b,c,d and experience outage due to the traffic demand in the vehicles. Figure 4.5 shows a possible cell tower coverage adjustment once the outage or coverage hole has been detected. In the latter figure, cell 204b and 204c dedicated all their resources to cover the traffic jam while creating holes behind them, opposite to the high way; however, tower 204a realigns to support these cells. Tower 204g and new towers 204e,f join as well to increase the resources.

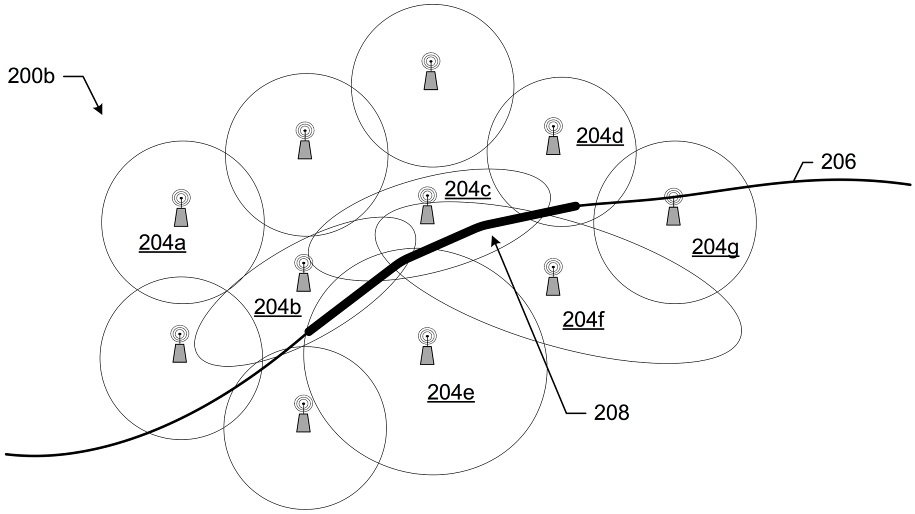


Figure 4.5: When outage is reported, neighboring cells may dedicate some of their resources to support cells experiencing outage.

The cell tower adjustment leverages the inter-RAT measurements on the UEs, activity that they perform regardless anyway. The main advantage is that the solution is temporal, fast, adjustable, and cost effective; for example, there is no need to install another tower, that would require property rental costs and more equipment. Also, traffic jams may occur anywhere along the street and major highways; having a solution that is fast and reactive is desirable.

4.3 Relevance to the Thesis

US Patents 9660710, 8868067 [25] and 9538413, 9154978, 8995255, and 9363727 [24, 23] is a collaborative effort with my Intel manager Joey Chou. This work focuses on measuring a wireless network, specifically cellular LTE or E-UTRA as it is formally known, in order to improve the system performance. When a cellular tower suffers outage or discovers a coverage hole, a corrective action must take place; these patents propose a method to measure, discover, and correct such anomalies. This thesis aims at providing service continuity; in the LTE networks, the solution to a network outage or coverage hole maintains the LTE network in operation, reliable for the mobile devices to enjoy service continuity.

CHAPTER 5

Vehicle-to-Infrastructure Experiments

5.1 Introduction

Current smartphone devices are equipped with at least two wireless interfaces: WiFi and cellular LTE. Cellular improvements on LTE rival the performance benefits of WiFi networks nowadays, but LTE carriers implement capacity limits onto their users with heavy financial penalties or service throttling. Relying solely on LTE connectivity, therefore, may become prohibitively expensive. In this heterogeneous environment, it is desirable to maximize the use of WiFi coverage when available before a connection moves over to cellular; that is, maximize the cellular traffic offload. Multipath TCP [43] is a backward-compatible evolution of TCP that allows precisely that without breaking the end-to-end connection. MPTCP performs seamless network handovers while devices maintain their connection sessions undisturbed; this is achieved in a *make-before-break* manner. Implementations exist today in Apple iOS®[6] in the Linux kernel [96].

A recently released Cisco VNI forecast [103] predicts that by 2020 traffic from wireless and mobile devices will account for two-thirds of the total IP traffic; smartphones alone will account for 30% of the traffic. Video traffic will take the lion's share with 82%, with content-delivery networks carrying nearly three-fourths of all the traffic. These trends respond, in part, to an increase in device capabilities and network bandwidth of cellular providers. To cope with this kind of traffic demand, MPTCP is a practical, backward compatible solution for mobile devices. Some contributions [20, 30, 7] already measure the performance of MPTCP, but they do it under controlled scenarios or via simulation.

In this chapter we present a thorough study of MPTCP performance in real deployment scenarios: a VANET scenario. The VANET is a wireless network that depicts high, predictable mobility.

Among its goals are the mitigation of vehicular traffic congestion, the assistance of drivers in danger, and the delivery of content. With the expansion of WiFi metropolitan networks, like the Cable WiFi Alliance[1] initiative, vehicles and pedestrians can offload cellular traffic to WiFi hot spots using MPTCP. The following is a summary of our findings:

1. MPTCP is capable to leverage WiFi networks on vehicles moving at velocities up to 40mph, 64km/h, or less.

2. MPTCP offloads between $25 - 50\%$ traffic off the cellular path, with best case scenarios at up to 60%.

3. Once the hardware is able to detect and associate with a local AP, MPTCP is able to establish a TCP subflow within a data packet RTT.

4. There is a trade-off between traffic offload and the download time; it may take MPTCP up to twice the time, in the worst case, to download a file compared to single-path TCP.

5. Long MPTCP flows are more vulnerable to the highly mobile dynamics than short or HTTP flows, which is the fundamental reason of the trade-off before.

6. MPTCP is not suitable to applications with tight latency constraints, like live video conferencing; however, we do not expect any issue with video transmissions that allow buffering.

5.2 Vehicular Test Bed

In this section we describe the vehicular MPTCP test bed used in our experiments. We describe the hardware and the relevant performance parameters used, as well as some middlebox issues with MPTCP connections found in our Department's network.

[1] See http://www.cablewifi.com for further details

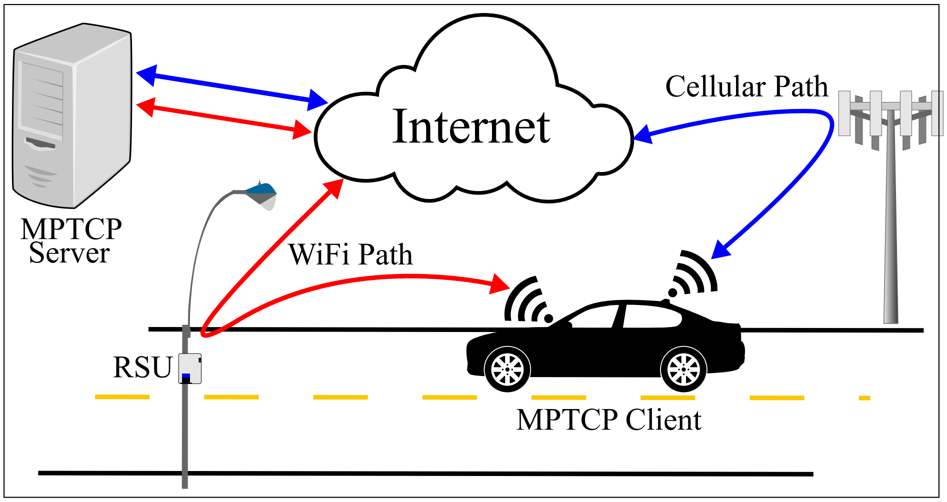


Figure 5.1: Vehicle-to-Infrastructure configuration

5.2.1 Hardware

We investigated the performance of MPTCP on a mobile test bed representative of the VANET scenario described in Section 2.1, Vehicle-to-Infrastructure, (V2I), shown in Figure 5.1. The V2I configuration in our test bed consists of a mobile *MPTCP Client* that connects to a remote *MPTCP Server* through the Internet via a fixed, roadside *access points*, APs, along the vehicle's path. The server is a commodity Intel PC running Ubuntu 14.04 LTS and MPTCP v0.90, equipped with a 1Gbps Ethernet card connected to an ISP fiber-optic service provided by Verizon FIOS® (now Frontier Communications); to see the connection performance, see Table 5.3. The server runs a web server, Apache, using the well-known ports 80 and 443, but we use port 5280 to separate our measurements from normal traffic. The servers uploads traffic as described in Section 5.3.1. The mobile client is a Lenovo Thinkpad® W520 laptop that also runs Ubuntu 14.04 LTS and MPTCP v0.90. It connects to the test bed via two interfaces: cellular and WiFi. The cellular connection is provided by an Apple iPhone® 5S via a USB cable; the phone establishes a LAN with the laptop through the USB cable in order to eliminate any wireless interference. The laptop also connects to the WiFi network via a TP-Link TL-WN722N® USB dongle, but we replaced the detachable antenna with an external, 9dBi antenna mounted outside the vehicle for better signal reception. Our test bed includes two implementations for the APs; namely, a WiFi *Roadside Unit*, RSU, and a commercial metropolitan WiFi network provided by Time Warner Cable, TWCWiFi®, part of the Cable WiFi alliance; we will refer to this network by its commercial name, TWCWiFi, throughout this text. The WiFi RSU is a TP-Link TL-WR841N® router running OpenWRT Barrier Breaker® (14.07) and it provides coverage on the 2.4GHz ISM band for IEEE 802.11n on channel 1 (2412MHz, 20MHz bandwidth) using a 5dBi dual antenna array with a theoretical maximum bit rate of 300Mbps. The RSU is a WLAN (private IP addresses) that provides Internet access to its clients via a Time Warner Cable Internet cable line with an advertised bandwidth of 100Mbps; for performance measurements, see Table 5.3.

5.2.2 Relevant MPTCP Parameters

Our test bed adopts many settings used in [20]; we repeat them here for completeness. Unless otherwise stated on this text, we use the default Linux parameters as set on [96].

Initial Congestion Window. As discussed in [35], we start every TCP subflow with an initial window size of 10 in order to unnecessarily hurt short-loved transmissions.

Slow Start Threshold. The default ssthresh in the Linux implementation [96] is set to "infinite" in TCP New Reno and zero when TCP BIC [127] and TCP Cubic [50] are used. These settings may adversely affect the performance of short-lived flows to the same destination (when metric caching is used). An initial (or cached) low ssthresh value causes the flow to enter into congestion avoidance sooner, where the congestion window grows slower, thus unnecessarily slowing the completion of the flow [20]. A large ssthresh value on a path with little loss, such as a cellular connection experiencing bufferbloating [42], would cause the RTT of the path to grow significantly, thus affecting the performance of the MPTCP scheduler. To be fair, we adopted a ssthresh value of 64KB.

Metric Caching. Linux kernels 2.6 and newer cache several TCP metrics for closed connection so that new ones in the future may use these settings as they connect to the same destination. This may be detrimental in some scenarios as shown in [54]; we do not cache metrics in our experiments.

Receive Buffer Size. As MPTCP works with multiple TCP subflows, it needs a larger buffer space to put the arriving segments in order. If the subflows depict asymmetric characteristics, the performance of the MPTCP session may degrade while attempting to reorder the segments [7]. We increase the received buffer space to 8MB.

Other TCP Parameters. Parameters like SACKs, Forward ACKs, Windows Scaling, and Timestamps for all our TCP subflows keep their default values in our test bed. Unless otherwise noted, we used LIA for MPTCP and TCP New Reno for traditional single-path TCP, SP-TCP. connections.

5.2.3 Issues with Middleboxes

We experienced the issue of middleboxes removing the MPTCP options from our TCP packets passing through the cellular networks (AT&T, Verizon, and T-Mobile) as in [20]. We changed our remote servers to use port 5280 and the issue was resolved.

We run an identical web service configuration on our CS Department server (CS Server) but we could not establish MPTCP connectivity with the mobile client even if we use a non-reserved port. After capturing some traces with *tcpdump* and analyzing them on *Wireshark*, we discovered that our CS Department Router (CS Router) overwrites the MPTCP entries in the TCP header with NOPs. In Table 5.1 we show an example of a departing SYN packet from one of our MPTCP Clients carrying an MPTCP capable option set and a checksum (12 octets); note that twelve NOPs overwrite completely and exactly this TCP option after the SYN MPTCP capable arrives at the CS Server. The distance between the CS Router and the CS Server is one hop connected through a switch. We also performed similar connections using other networks around the our campus and we did not encounter this issue; therefore, we concluded that the misbehaving middlebox is our CS Router. To cope with this issue, we moved our MPTCP server to a fiber-optic link (Verizon FIOS service) where no TCP options are blocked, until our department routers are properly configured. Table 5.3 shows the optical link upload speeds used by the MPTCP server; note that the Verizon FIOS upload speed exceeds all other download speeds at any access point available to the mobile client, for either or both WiFi, cellular, or combination of both; therefore,we consider that our decision to use this optical link does not constitute a performance compromise nor it is a bottleneck in our experiments.

Host	TCP Options
CS Server	[mss 1380, sackOK, TS val 2220966 ecr 0, nop, wscale 9, **nop**,**nop**,**nop**,**nop**, **nop**,**nop**,**nop**,**nop**, **nop**,**nop**,**nop**,**nop**]
MPTCP Client	[mss 1460, sackOK, TS val 2220966 ecr 0, nop, wscale 9, **mptcp capable csum 0x3f6b1cd886723e46**]

Table 5.1: MPTCP issues with middle-boxes

5.3 Methodology

Our experiments consist on downloading repeatedly files of distinct sizes from our MPTCP Server, see Table 5.2. For the traffic labeled as HTTP in Section 5.3.1, our scripts download, unless otherwise noted, 100MB of data per experiment. Small files download with a configurable probability ρ, which represent web objects, while large files download with a probability $1 - \rho$; we configured $\rho = 0.80$ to model the traffic as described in Section 5.3.1.3. We also download repeatedly single files as a single experiments in order to observe longer the performance of MTPCP. Whenever we perform one such experiment, we state the size of the file used; for these experiments, we run the experiment twenty times in order to limit temporal factors in the network.

To capture the measurements, we use *tcpdump*[2] on both the source and the destination of a connection under observation. Since the source host in all our experiments is the MPTCP Server, the *pcap*[3] files that this host generates capture the correct RTT and packet loss of each experiment. The capture file at the mobile client gives us the throughput, the download time, the traffic offload for each path, as well as the overall MPTCP out-of-order delay in the connection. To obtain these metrics, we analyzed each captured file with the tool *tcptrace*[4]. In all our experiments,

[2] Describes the packets on the network interface.

[3] Dumps or packet traces generated by tcpdump.

[4] Analyzes the pcap files that tcpdump generates.

Type	Repeated 20x	HTTP Traffic
Small	64KB, 512KB	[1,99]KB, 1KB incr [100,900]KB, 100KB incr
Large	4MB, 32MB, 500MB	100MB, 200MB, 500MB

Table 5.2: File sizes used in our traffic generation

we maintained a constant pinging to the MTPCP Server in order to maintain our cellular devices registered on the RRC at the base station; otherwise, the cellular Tower may forget our devices and introduce extra delay in our experiments.

5.3.1 Traffic Generation

Mobile clients establish MPTCP connections with the remote server and perform HTTP requests over a list of available files of distinct sizes shown in Table 5.2. We describe next how files are selected and how traffic is generated.

5.3.1.1 Short Flows

MPTCP clients establish flows that download files of size 512KB or smaller; these correspond to the *small* category in Table 5.2. Short-lived HTTP connections are particularly wasteful of network resources due to the high ratio between overhead to actual data transmission [115]. We use these file transfers to understand their impact on the MPTCP performance under mobility.

5.3.1.2 Long Flows

We also generate traffic using long flows, rather than single or multiple short flows as described in Section 5.3.1.1 and Section 5.3.1.3 respectively. Long flows correspond to 4MB, 32MB and 500MB file transfers and they aim to explore the performance on MPTCP over long, sustained transmissions during path disconnections and reconnections due to mobility.

5.3.1.3 HTTP Traffic

To better represent the traffic patterns of common HTTP sessions in the Internet, we model and automate the HTTP traffic generation rather than downloading fixed-sized files as in the previous sections. We create scripts that repeatedly download random small-sized files from the remote server in order to emulate the transfer of small HTML objects in a web page. For this, we used the Linux tool *curl*[5].

For our download experiments, our script prepares bursts size between one to four megabytes for data to be requested. Each burst is then broken into independent *curl* requests composed of small file sizes shown in Table 5.2; each of these requests represent a web object or an HTTP request, and all together form a burst that represent a web page. The data transfer stops when the entire burst is complete and another burst may start again. The time separation of the bursts follows a Poisson distribution with $\lambda = 10$ms; this value produces a continuous and sustained data traffic without becoming CBR traffic. For example, in our experiments we may say that we generated 100MB of HTTP traffic; that means that the MPTCP client downloaded 100MB worth of bursts, or web pages, from the MPTCP Server and that represent HTTP traffic. Finally, bursts may be transmitted in serial or in parallel; in our experiments, we used serial transfers in order to maximize the cellular LTE providers' data caps and minimize cost.

5.3.2 Performance Metrics

We used the following performance metrics in our experiments:

Download Time. Time spent between the SYN packet of the HTTP request until the last packet data packet it received.

Throughput. Amount of bytes delivered per unit of time, including MPTCP overhead and lower layer headers.

Traffic Offload Ratio. The transmitted data ratio between the two paths used in a MPTCP connection.

[5] Downloads data from the server without user interaction.

RTT. The round-trip time is a TCP metric that measures the time it takes to acknowledge the receipt of a data segment.

Out-of-order Delay. Time spend in ordering the arriving data stream from all TCP subflows and delivering to the upper layers.

Packet Loss. The ratio between the retransmitted packets and the total packets sent in a connection.

5.3.3 Experiment Scenarios

We use the following three basic scenarios to perform our MPTCP study.

V2I using WiFi RSU. In the V2I configuration, the MPTCP client drives around the coverage of an WiFi RSU at distinct velocities: 20, 30, and 40mph, roughly 32, 48, and 64km/h. The WiFi RSU is placed on a street sign pole at 2m from the floor, where there may be some trees and parked vehicles that may obstruct coverage. The vehicle loops around Venice Bl., periodically entering and leaving coverage, depending on traffic conditions such as street semaphores at intersections or random obstructing vehicles. The WiFi RSU connects to the Internet via a TWC Cable modem at a residence nearby; for connection speeds, see Figure 5.3.

V2I using TWCWiFi. Under this V2I configuration, the vehicle moves throughout the city of Los Angeles TWC WiFi metropolitan coverage at an approximate velocity of 30mph, or around 48km/h. The connection components of this use case are completely out of our control; that is, we only control the MPTCP client, the AP is under TWC domain. The vehicle drives around the streets of Los Angeles connecting and disconnecting from the available APs to our vehicle passing by. We provide an approximation of 30mph because the AP placement by Time Warner does not allows us to maintain a constant velocity. AP placement is not on the major streets; it is along alleys, low speed roads or places with high concentration of people, i.e., schools and parks.

Due to safety concerns, in all our experiments we do not operate our vehicles faster than any posted speed sign; moreover, all statistics collection is automated so our drivers are fully aware of the traffic conditions at all times. On each run, we perform the following experiments: (1) 20x downloads of file sizes of 64KB, 512KB, 4MB and 32MB; (2) 1x 100MB of HTTP traffic; (3) 1x

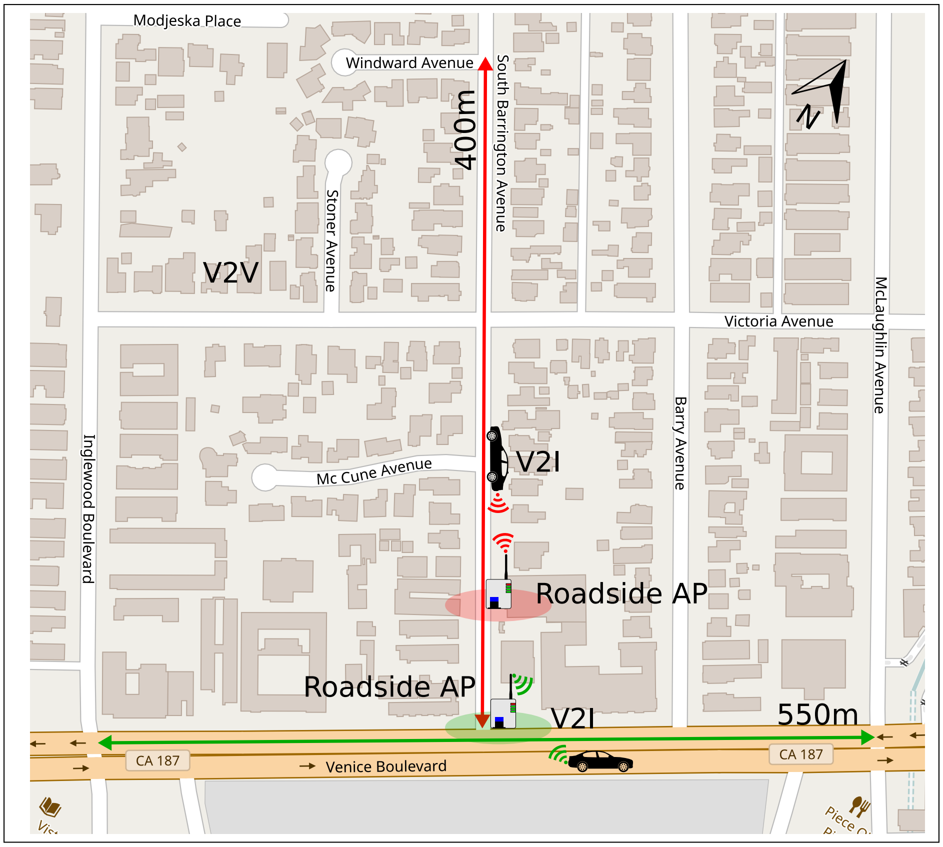


Figure 5.2: Routes for WiFi RSU road experiments

500MB file download.

5.3.3.1 Vehicle Routes

For our V2I experiments, our test bed has a WiFi RSU placed at two distinct locations close to a wired Internet gateway; see Figure 5.2 for the RSU locations. Additionally, it also uses the TWC WiFi network for completeness; see Figure 5.3 for the network layout.

Figure 5.2 shows the main vehicle routes used in our experiments as well as the locations of our WiFi RSUs. We used two locations for WiFi RSU due to the legal posted speed limitation. The first location is on Barrington Ave., about 100m from Venice Bl.; this placements serves for the 20mph and 30mph experiments and the route trajectory is shown in red. The second location is on the corner of Venice Bl. and Barrington Ave., and the vehicle route for cars moving at 40mph is shown in green. For the red routes, the path is 400m long and it allows the safe operation of vehicles at

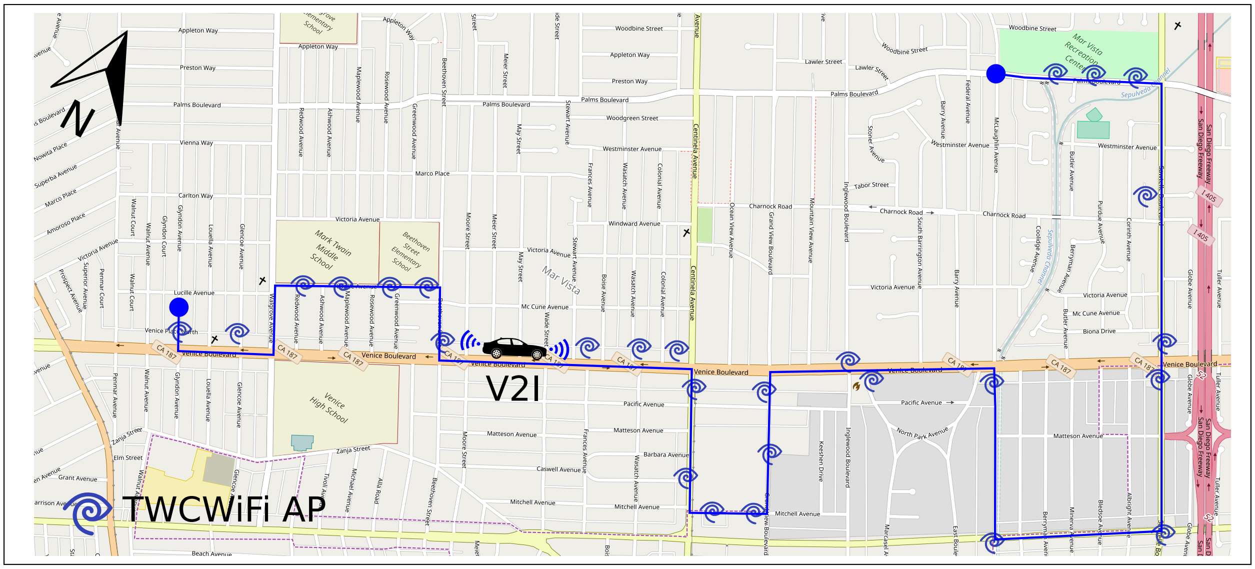


Figure 5.3: Routes for TWCWiFi road experiments

small velocities; moreover, it includes several objects that may induce some wireless interference, i.e., parked and passing vehicles, trees, electric posts, etc., and a traffic intersection with a stop sign. Our WiFi RSU for this route is shown in red as well. For our experiments at 40mph, we use the 1100m green route in Figure 5.2, with the green WiFi RSU on Venice Bl. (California Highway CA187). Our vehicle looped on CA187 around the Inglewood Bl. and McLaughin Bl. intersections, where the vehicle stop momentarily. The route characteristics are about the same as the red route, except that vehicles move at the posted speed limit of 40mph and there are two semaphores, instead of one stop sign.

We also perform experiments using an independent network currently available on the streets of Los Angeles: the TWCWiFi, part of the Cable WiFi Alliance. Figure 5.3 shows the complete route. This route is designed to maintain the test bed vehicle covered by a WiFi network as much as possible. It includes major streets, such as Venice Bl. and Sawtelle Bl., with a large number of vehicles and traffic semaphores as well as small, residential streets with stop signs and pedestrian traffic. Under these conditions it is difficult to maintain a constant velocity like in the RSU experiments; however, the average velocity was 30mph that ranged from a maximum of 45mph to a complete stop at traffic intersections. This route include large traffic diversity conditions; therefore, this experiment is as real as an MPTCP user would experience under the same conditions.

Wired Internet Gateways

Carrier	RTT (ms)	Download (Mbps)	Upload (Mbps)
Verizon Fios	9.30 ± 0.82	58.50 ± 0.29	64.32 ± 0.06
CS Department	1.40 ± 0.52	588.55 ± 25.48	596.32 ± 3.90
Cable (TW Cable)	12.33 ± 1.03	117.00 ± 0.33	11.87 ± 0.04

Wireless Internet Gateways

Carrier	RTT (ms)	Download (Mbps)	Upload (Mbps)
WiFi (TW Cable RSU)	13.70 ± 2.50	33.21 ± 4.24	12.11 ± 0.14
TWCWiFi (City WiFi)	14.10 ± 0.32	36.63 ± 12.45	5.87 ± 0.08

Cellular Internet Gateways

Carrier	RTT (ms)	Download (Mbps)	Upload (Mbps)
AT&T	34.80 ± 2.97	27.45 ± 1.50	3.86 ± 1.18
T-Mobile	49.30 ± 2.63	5.80 ± 1.55	1.10 ± 0.49
Verizon	37.20 ± 4.49	7.38 ± 2.41	1.17 ± 1.27

Table 5.3: Speedtest® measurements of Internet gateways

5.3.4 Number of Paths

We setup our test bed to establish end-to-end connectivity using the following multipath configuration:

Single-Path TCP. Baseline configuration used for MPTCP comparison. Unless otherwise noted, we used Cellular AT&T as the single-path TCP.

2-Path MPTCP. Cellular and WiFi paths available to the mobile MPTCP client; see Figure 5.1.

5.4 Experiment Results

This section presents our results on the MPTCP performance when it operates under mobility in Vehicle-to-Infrastructure (V2I) configuration: *a mobile, end-to-end connection between two MPTCP-capable hosts over the Internet connected wirelessly via cellular and WiFi paths on the mobile side, and wired on the server*. Under this scenario, the current state-of-the-art reference is single-path TCP, SP-TCP, operating on a cellular connection; our goal is to study the feasibility and efficiency of MPTCP at leveraging both a cellular and WiFi coverage under mobility. In all our experiments, we use AT&T for cellular coverage. For WiFi Internet access, we use a

WiFi RSU and TWCWiFi as described in Section 5.3.3 and follow the V2I routes described in Section 5.3.3.1 at 20, 30, and 40mph; higher velocities allowed little connection time to be relevant. TWCWiFi experiments operate at approximately 30mph with high variations including stops and accelerations. We performed these experiments over the months of May through December 2016; the metrics used are download time of fixed-size file transfers, the throughput recorded, the traffic offload onto the WiFi path, and the latency measurements, including round-trip time and out-of-order delay. We present our results as a function of the vehicle velocities for both small and large flows, very long continuous flows and HTTP traffic; they are described in Section 5.3.1.

5.4.1 Download Time and Throughput

In this section we present both the download time and the throughput performance observed in our test bed experiments. We first provide a small discussion about the Internet measurements for the ISPs used. Then we continue with our presentation of our test bed findings in the following order: first on small file transfers (64KB and 512KB); then large transfers (4MB and 32MB); and finally continuous transfers (500MB and 100MB worth of HTTP traffic).

5.4.1.1 Speedtest Measurements

Before we discuss our test bed findings, we present our observations of the ISPs used in our test bed; they are summarized in Table 5.3. To obtain these measurements, we use a third-party reputable measurement tool available freely online called Speedtest®. This table groups the Internet access in *wired*, *wireless*, and *cellular*. For the wired Internet access, we used Verizon Fios® (now Frontier Communications), our university ISP, and Time Warner Cable®, (TWC, now Spectrum). For our wireless Internet APs, the WiFi RSU and the TWCWiFi, we used the wired ISP Time Warner Cable network described before; however, for WiFi RSU, we used a residential cable service while TWCWiFi uses the metropolitan available network of TWC. Finally for our cellular Internet access, we use three cellular carriers: AT&T, T-Mobile, and Verizon.

Our table summarizes the network characteristics, however, we point out the issue with middleboxes described in Section 5.2.3. Due to our CS department outdated routers at the time of the

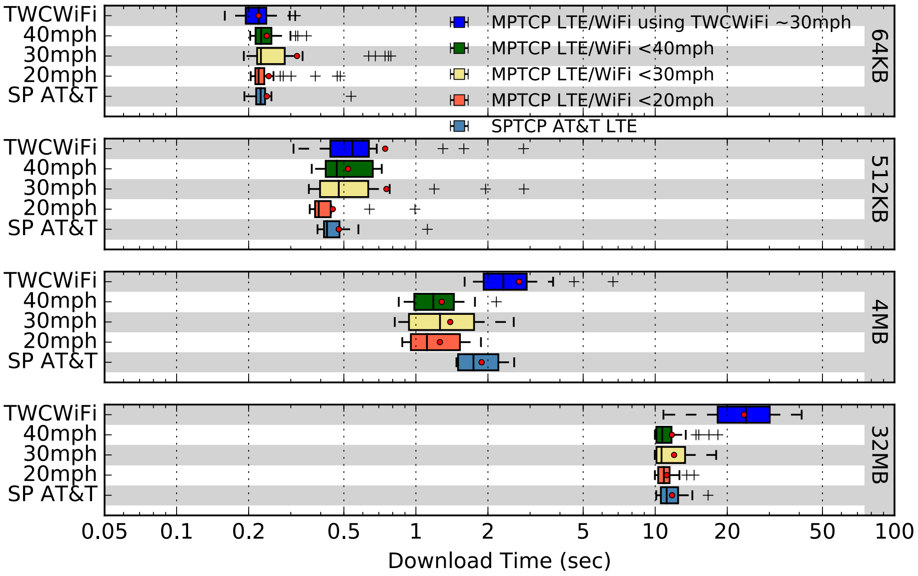


Figure 5.4: Compilation of V2I download time experiments

experiments, we could not establish MPTCP connection. Therefore, changed our MPTCP server's connection to a fiber-optic network provided by Verizon Fios. Note that the performance of the fiber-optic path, in both upload and download speeds is larger than any of the other Internet access used in our test bed by our moving vehicle; except of course, our CS department network and the TWC Cable modem. We do not considered this to be any issue nor does it impact our results in any way, since there is only one mobile MPTCP client that our server responds in our test bed.

5.4.1.2 Small Flows

Our download time results are summarized in Figure 5.4; we use single-path AT&T as the current state-of-the-art reference since it is the best performing cellular connection in our experiments, compared to T-Mobile and Verizon. The 64KB and 512KB are considered the small flows in our experiments. From the figure we can observe that there is no significant impact on the download performance at any vehicle velocity when the connection operates using either MPTCP or SP-TCP. In particular, 64KB flows show almost identical download performance compared to SP-TCP. The reason is evident when we consider that the *ssthresh* setting in Section 5.2.2 in our experiments is 64KB; this means that the transmission occurs mostly under the Slow Start phase and therefore, neither the MPTCP's nor the TCP's Congestion Avoidance stage is triggered. Moreover, the ve-

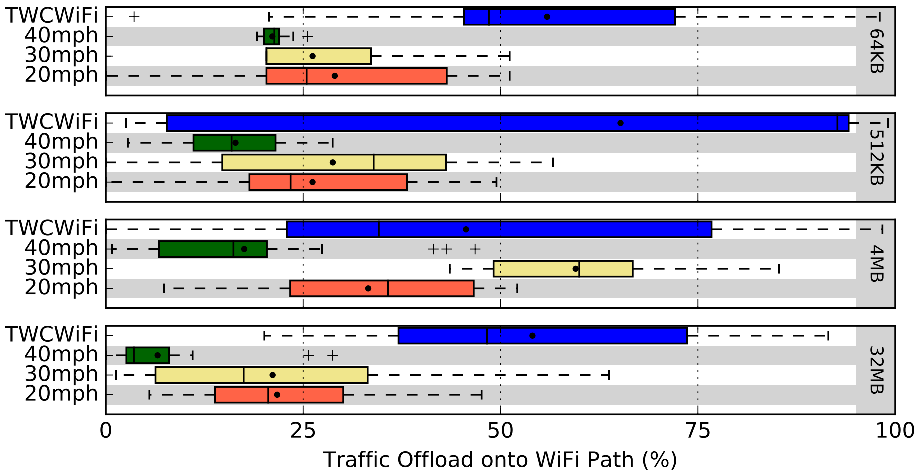


Figure 5.5: Compilation of V2I cellular traffic offload onto the WiFi path

hicle's velocity is not a factor because given the small flow lifetimes and sizes and the throughput measured for the connections, Table 5.3, the file download time is very short and thus we have small vehicle displacement: 3.5m for around 200ms in 64KB transfers and 9m for around 500ms in 512KB transfers; both at the worst case velocity of 40mph. Despite the short time, MPTCP is still able to negotiate another TCP subflow with the remote server within one RTT on the WiFi Path, as shown in Table 5.4; compere the delay times from this table to those in Table 5.5. Figure 5.5 shows the percentage of traffic offload measured. For 64KB flows, MPTCP is able to offload between 21% to 29% when using WiFi RSU and up to 55% when using TWCWiFi; for 512KB, MPTCP offloads between 16% to 28% in WiFi RSU and 65% when using TWCWiFi. For small flows, we see that velocity does indeed play a role in the amount of offload percentage measured; however, we observed that the TWCWiFi has much better offloading results. We attributed this to the placement and visibility of the AP compared to the WiFi RSU; that is, TWC places their APs on their wiring infrastructure at highly visible locations while we placed our RSUs on the sidewalk at 2m height.

5.4.1.3 Large Flows

Figure 5.4 also shows the 4MB and 32MB flows, which we consider as large flows. There are two interesting findings that we observe for large flows: 1) TWCWiFi network has worse perfor-

MPTCP WiFi Path setup delay (ms)

Speed	64KB	512KB	4MB	32MB	500MB	HTTP
20mph	85.26 ± 16.01	83.52 ± 8.31	94.94 ± 17.03	91.45 ± 9.51	104.26	88.21 ± 23.52
30mph	93.04 ± 29.59	88.56 ± 10.08	87.99 ± 11.90	96.42 ± 27.32	84.29	93.46 ± 49.52
40mph	85.26 ± 17.53	83.52 ± 13.69	94.94 ± 23.82	91.45 ± 245.60	104.26	92.30 ± 26.12
TWCWiFi	85.26 ± 40.65	83.52 ± 18.31	94.94 ± 81.44	91.45 ± 52.04	104.26	84.66 ± 38.63

Table 5.4: V2I average delay to establish another TCP subflow once a new path has been discovered

mance than either SP-TCP or WiFi RSU at an average velocity of 30mph; and 2) MPTCP under WiFi RSU either beats or matches SP-TCP. For TWCWiFi connections in our first observation, the download times for 4MB flows take $2x$ (twice) the time compared to WiFi RSU and almost $1.5x$ that of the SP-TCP; for 32MB, TWCWiFi takes twice the time to download the file compared to WiFi RSU and SP-TCP. However, when we compare WiFi RSU performance to that of SP-TCP, we observed that MPTCP either beats or matches the cellular path's performance, that's our second finding. Given these two observations, we attributed the underperformance of TWCWiFi the fact that large flows are exposed longer to the vehicle dynamic conditions: stop-and-go, vehicles, hoping to distinct routers, etc; see Table 5.5 and notice the higher RTT and loss figures for 4MB and 32MB compared to small flows. However, Figure 5.5 shows that TWCWiFi has much better traffic offloading than WiFi RSU, particularly for 32MB flows; for WiFi RSU we got 37% and 16% for 4MB and 32MB flows, and for TWCWiFi we observed 45% and 54%, respectively. These results show that long flows are more exposed to traffic conditions and as a result that adversely affects the download performance by up to twice the download time in the worst case. However, we also observe high offloading under either vehicle's velocity. In other words, MPTCP is able to offload the cellular path at any particular velocity, with a trade-off in download time performance.

5.4.1.4 HTTP Traffic and Continuous Flows

Figure 5.6 shows the performance of both long, continuous flows, 500MB, and HTTP traffic. We could not perform live video conference experiments or other video rendering applications while driving due to security concerns. Considering that video transfers are commonly buffered (e.g. HTML5), we assume that video traffic resembles that of a large or a very large traffic flow. Starting with our 500MB download, we observe the same trend described in Section 5.4.1.3 for

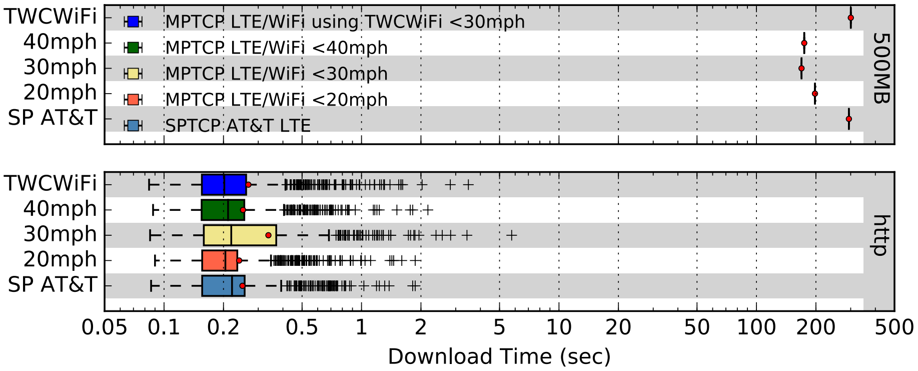


Figure 5.6: Download time for V2I HTTP traffic and long, continuous flows

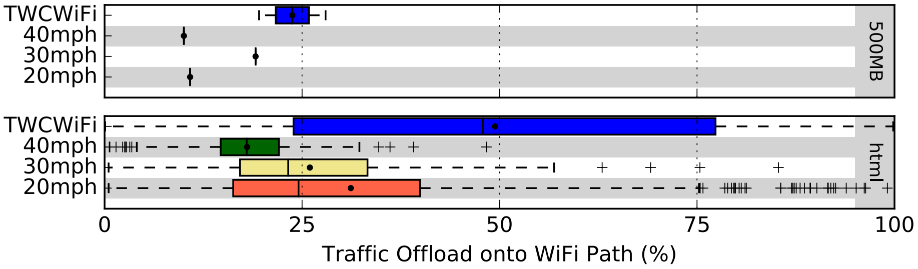


Figure 5.7: Traffic offload for V2I HTTP traffic and long, continuous flows

large flows: in the long run, MPTCP either beats or matches SP-TCP in download times. From the figure, WiFi RSU completes the download $1.6x$ faster than SP-TCP for any speed; while TWCWiFi takes just about 2.2% more time to download the large file than SP-TCP. In contrast, Figure 5.7 shows TWCWiFi offloading just less than 24% of the cellular traffic, while WiFi RSU offloads about 13% of its cellular traffic. These results corroborate that extremely large flows are more vulnerable to traffic conditions and network disconnections; however, MPTCP sessions recover fast but at a penalty of increasing the download time.

HTTP traffic shows a different trend compared to long, continuous flows, as shown in Figure 5.6. Considering the file selection for download in Section 5.3.1 where small files constitute 80% of the traffic, we found no impact on the vehicle's velocity for either method used: MPTCP or SP-TCP. In general, all methods download 100MB worth of HTTP traffic in about 270ms, 340ms being the worst case for 30mph vehicle velocity experiments. This result demonstrates that MPTCP is

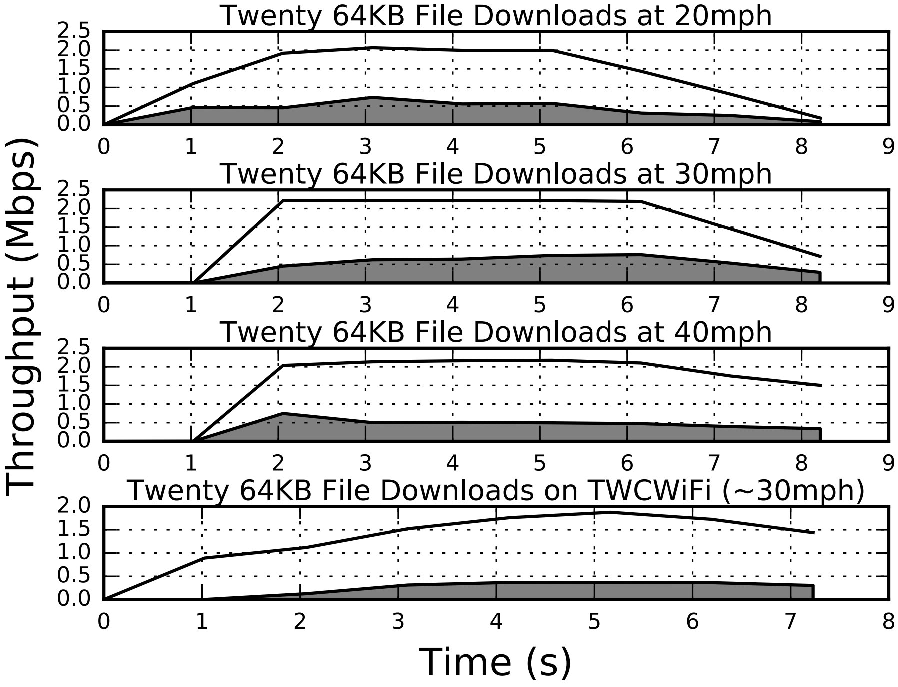


Figure 5.8: Throughput observed for 64KB flows at varying speeds

capable to achieve similar performance results to SP-TCP under a cellular path. Note that for HTTP traffic, 20% of the files transferred are files between 1MB to 4MB; thus practically removing the long flows. The advantage of MPTCP over SP-TCP appears in its capacity to offload the HTTP traffic. Figure 5.7 shows that TWCWiFi is able to offload just less than 50% of the cellular traffic while WiFi RSU offloads just less than 25%.

5.4.1.5 Throughput Observations

To complement our download time measurements, we provide a visual representation of the overall cumulative MPTCP throughput from all the TCP subflows that participated during the transmission at each experiment. In the throughput figures, the shaded area represents the WiFi contribution to throughput used while the line represents the overall throughput obtained throughout the transmission; thus, the blanked area in between corresponds to the cellular traffic.

Our first observation is a continuous, stable WiFi offloading of the cellular traffic on the small flows; see Figures 5.8 and 5.9. This shows that the time MPTCP takes to establish a TCP subflow

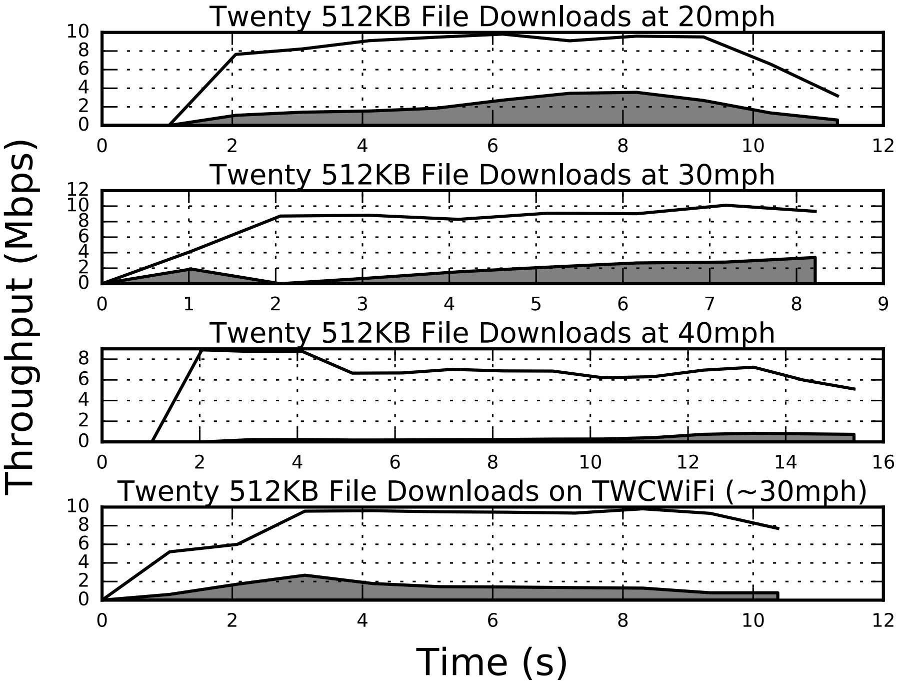


Figure 5.9: Throughput observed for 512KB flows at varying speeds

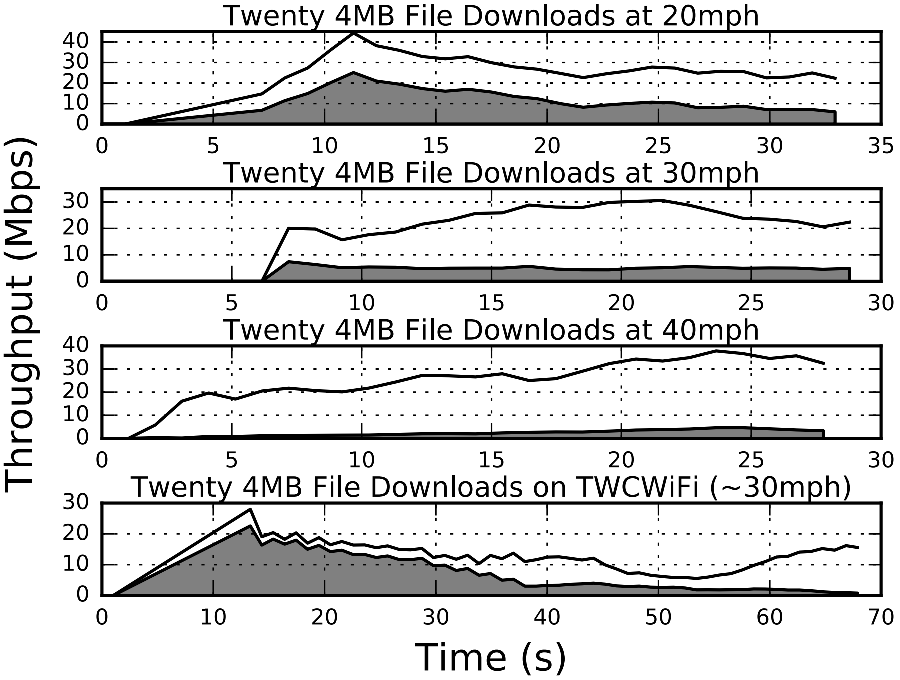


Figure 5.10: Throughput observed for 4MB flows at varying speeds

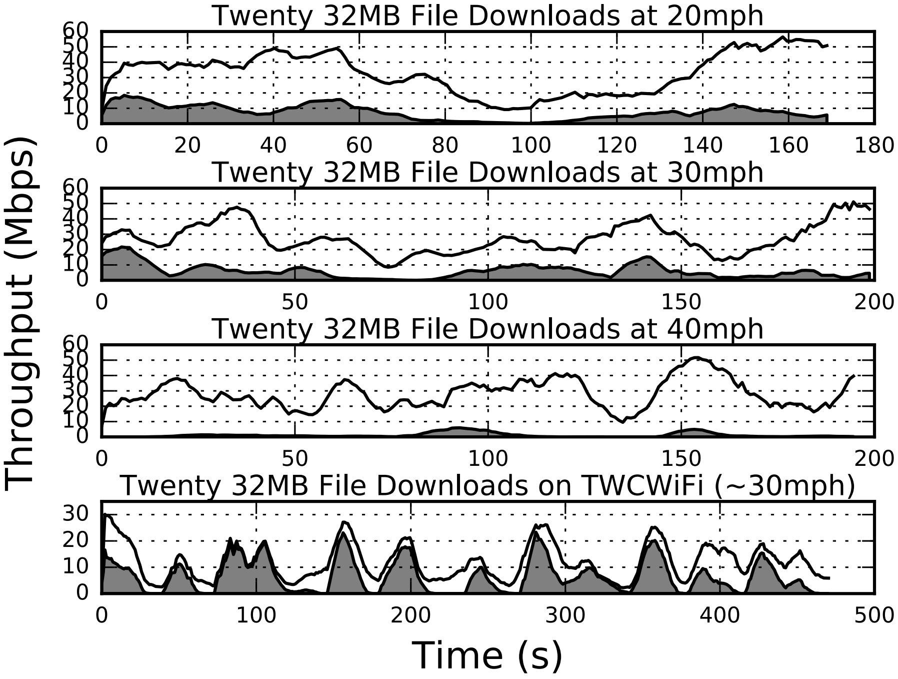


Figure 5.11: Throughput observed for 32MB flows at varying speeds

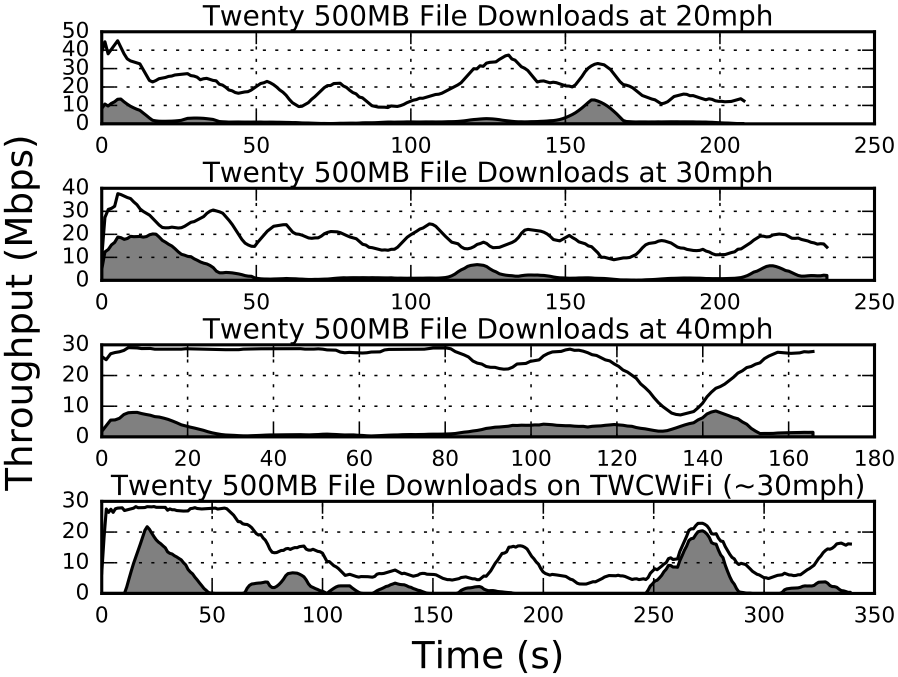


Figure 5.12: Throughput observed for 500MB flows at varying speeds

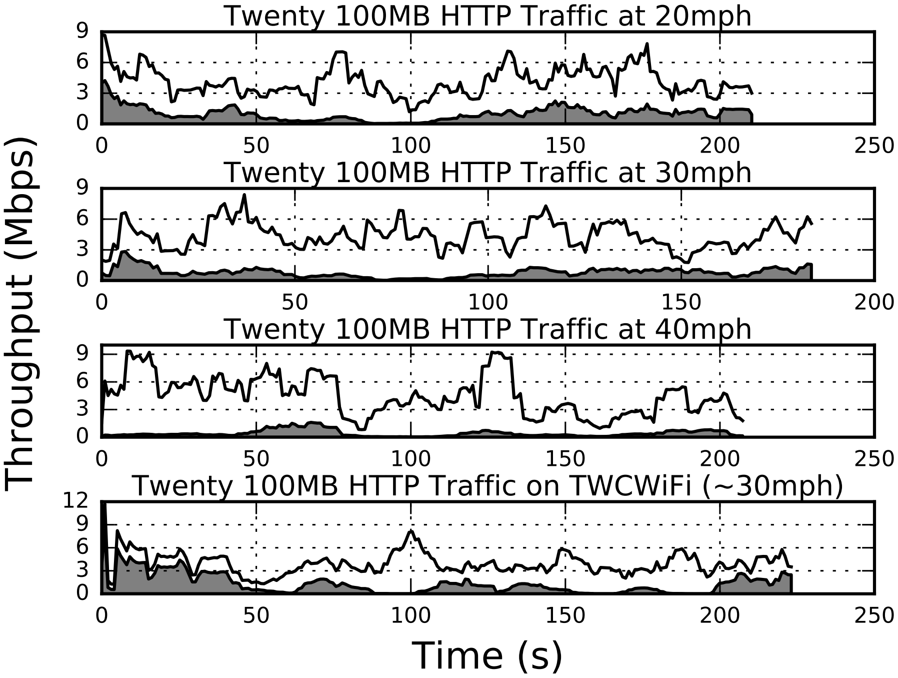


Figure 5.13: Throughput observed for HTTP flows at varying speeds

is negligible as it is unnoticeable during the transmission. Also note that these figures show twenty downloads of the particular file, therefore, MPTCP is able to offload traffic once it discovers and connects to a WiFi AP in the vicinity. A notable exception is flow 512KB at 40mph; this results shows little activity on the WiFi path, perhaps because the vehicle approached or left the WiFi RSU toward the end or beginning of this running experiment, respectively, and there was little chance to take advantage of the AP.

For large flows, we observe larger WiFi contributions towards the overall MPTCP throughput; this result demonstrates with higher visibility the most important advantage of using MPTCP over SP-TCP. 4MB flows show mixed results but overall they all demonstrate some offloading performed; see Figure 5.10. Also, we note that TWCWiFi offers larger WiFi contributions than our WiFi RSU; the reason is because the vehicle has a higher chance of WiFi connectivity as there are more APs, compared to the single WiFi RSU used. This is better shown in the 32MB flow under TWCWiFi; see Figure 5.11. Note how the mobile client discovered more than ten distinct APs while following the route described in Figure 5.3; also note that TWCWiFi highly contributes to the overall MPTCP throughput in the connection. In general, the WiFi RSU connections contribute

Cellular RTT(ms)

Speed	64KB	512KB	4MB	32MB
20mph	53.70 ± 11.84	69.02 ± 21.42	249.09 ± 139.45	169.12 ± 92.26
30mph	58.26 ± 5.27	70.43 ± 19.06	257.72 ± 148.00	167.15 ± 69.15
40mph	49.05 ± 4.92	65.72 ± 17.39	219.35 ± 120.34	196.47 ± 72.13
TWCWiFi	58.42 ± 10.26	106.69 ± 45.48	176.37 ± 97.79	114.75 ± 74.92

WiFi RTT (ms)

Speed	64KB	512KB	4MB	32MB
20mph	106.54 ± 55.96	134.33 ± 73.21	233.17 ± 143.04	185.15 ± 119.04
30mph	122.44 ± 58.93	182.63 ± 97.50	246.16 ± 138.72	227.64 ± 115.39
40mph	97.82 ± 40.99	140.10 ± 75.20	277.33 ± 185.46	288.38 ± 153.19
TWCWiFi	29.75 ± 6.04	85.43 ± 41.05	203.49 ± 123.88	127.34 ± 87.50

Cellular Loss (%)

Speed	64KB	512KB	4MB	32MB
20mph	0.21 ± 0.74	0.01 ± 0.06	4.87 ± 4.69	2.89 ± 2.11
30mph	0.07 ± 0.37	0.02 ± 0.07	1.87 ± 2.82	3.04 ± 2.47
40mph	–	–	0.99 ± 1.98	3.60 ± 2.31
TWCWiFi	–	0.54 ± 1.57	1.04 ± 1.84	0.93 ± 1.41

WiFi Loss (%)

Speed	64KB	512KB	4MB	32MB
20mph	1.39 ± 1.29	0.04 ± 0.10	4.82 ± 3.03	3.25 ± 2.02
30mph	2.05 ± 4.14	0.14 ± 0.35	2.59 ± 3.10	3.57 ± 3.36
40mph	0.44 ± 1.19	0.30 ± 0.43	2.15 ± 4.17	7.71 ± 16.20
TWCWiFi	0.22 ± 0.65	1.60 ± 3.14	7.92 ± 6.73	6.25 ± 5.39

Table 5.5: V2I RTT and Loss

a little less to the overall connection compared to TWCWiFi because the vehicle spends less time within its coverage. For further details on WiFi coverage see Section 5.3.3.1. Similar observations can be drawn from the 500MB flows and the HTTP traffic in Figures 5.12 and 5.13

5.4.2 Latency Analysis

In this section we analyze the performance of MPTCP focusing on latency. Latency aims to measure the response time to applications consuming data, such as video rendering or telephony applications. A use case where latency is more relevant than download time or throughput is a live conference call, for example. Due to safety reasons we did not perform such experiments in our test bed, however. We used our automated data transfers described in Section 5.3.1, provide our observations and draw conclusions.

	Cellular RTT (ms)		Cellular Loss (%)	
Speed	HTTP	500MB	HTTP	500MB
20mph	55.02 ± 10.26	84.00 ± 37.50	0.04 ± 0.37	0.16
30mph	59.78 ± 8.69	120.90 ± 60.40	0.04 ± 0.34	0.14
40mph	52.30 ± 7.69	90.00 ± 30.50	0.02 ± 0.42	0.01
TWCWiFi	65.27 ± 15.88	89.90 ± 49.20	0.08 ± 0.62	0.36 ± 0.40

	WiFi RTT (ms)		WiFi Loss (%)	
Speed	HTTP	500MB	HTTP	500MB
20mph	54.27 ± 12.94	128.00 ± 119.10	1.03 ± 3.49	2.44
30mph	68.09 ± 18.47	166.50 ± 104.40	1.77 ± 5.38	0.74
40mph	54.09 ± 13.16	82.20 ± 49.00	0.85 ± 3.58	0.21
TWCWiFi	34.71 ± 10.66	94.24 ± 77.60	0.75 ± 4.19	6.34 ± 9.44

Table 5.6: V2I RTT and Loss measurements for the HTTP traffic and long, continuous flows

5.4.2.1 Round-Trip Time

The Round-Trip Time, RTT, for both small and large flows in shown in Table 5.5, for long continuous flows and for HTTP traffic it is shown in Table 5.6. We provide a log-scale Complementary Cummulative Distribution of the RTT for all cellular transfers in Figure 5.14 and for all WiFi transfers in Figure 5.15. The CCDF figures show the probability that an RTT value is recorded with a particular delay time or longer, in milliseconds.

From the two tables we see that for general MPTCP connections, the RTT observed in WiFi transfers is consistently slightly larger than those in cellular paths; this is, of course, when there is a WiFi path available and when there is an ongoing transfer. Notice, however, that the WiFi path has larger loss than the cellular path, especially during large transfers. This shows that MPTPC large flows are expected to suffer larger degradation as exposed in Section 5.4.1.3.

Figures 5.14 and 5.15 show better the performance of MPTCP transfers in terms of latency. For small flows under the cellular path, we see that in general, 90% of the traffic experiences an RTT upper-bound of 100ms, except for 512KB TWCWiFi transfers that takes a little longer. On the WiFi path, that upper-bound delay experienced increased to 200ms for 90% of the observations; we have similar observations in HTTP traffic. Large file transfers, including the long, continuous 500MB exhibit a similar trend. For the cellular path, the RTT upper-bound is around 500ms for 90% of the traffic; for the WiFi path, we see a little more variation with upper-bounds also above

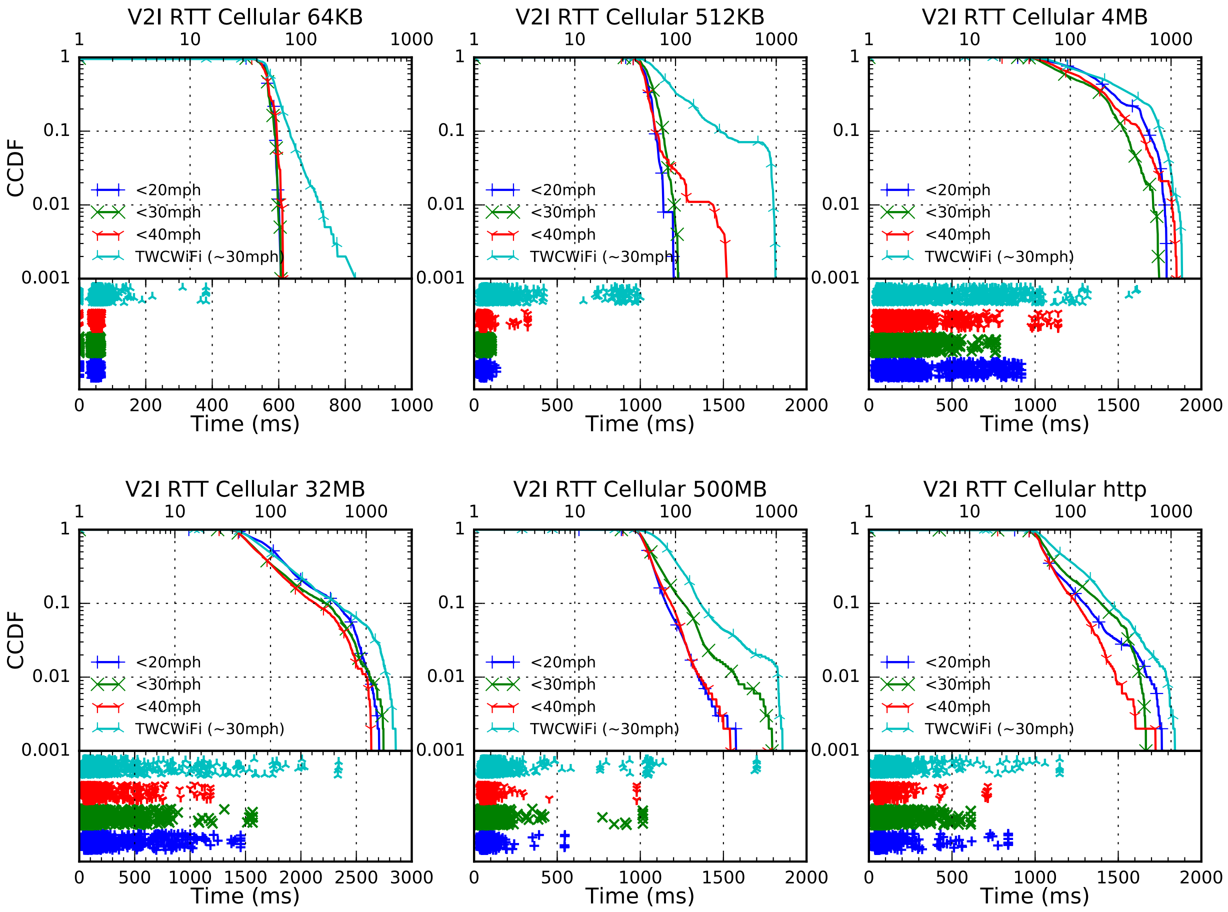


Figure 5.14: CCDF of the RTT on the cellular path

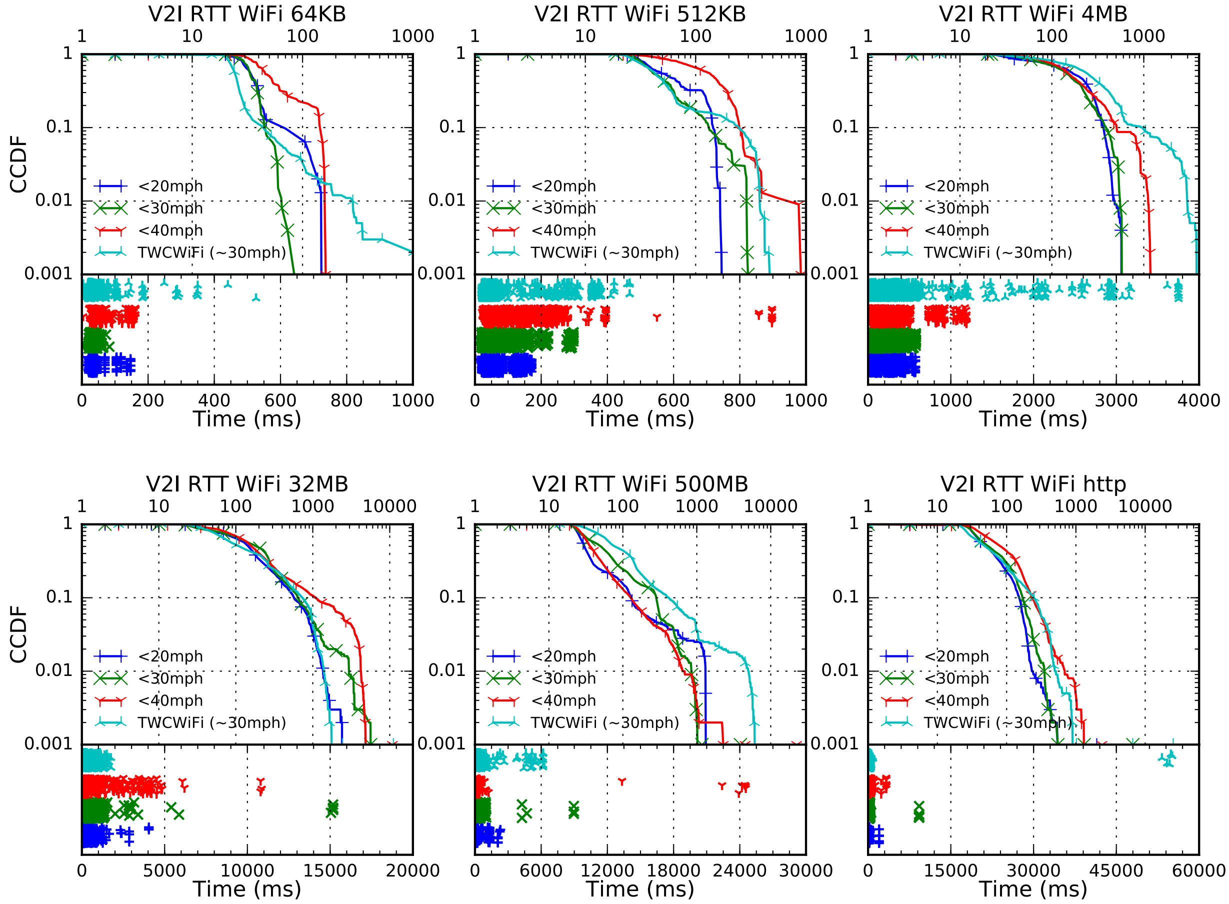


Figure 5.15: CCDF of the RTT on the WiFi path

the 500ms for 90% of the traffic. In the 32MB case for the WiFi path, that upper-bound is just shy of one second. These results show that mobility plays a significant factor on latency, under any velocity. However, our results show that both cellular and WiFi paths are similarity impacted; thus indicating that mobility is the principal culprit in this case, rather than MPTCP. In general, we expect that applications that have low latency requirements, such as live video conferences will experience significant performance impact under either SP-TCP or MPTCP; other video transfers where buffering is allowed may benefit from MPTCP due to the offloading performance and with buffering time larger than one second (the worst latency observed) in order to allow sufficient time to receive enough frames and provide a smooth playback experience.

5.4.2.2 Out-of-Order Delay

Out-of-order delay is an important measurement particular to MPTCP that helps us analyze how well MPTCP receiver reassembles the data stream fragmentation caused by the TCP subflows at the sending end. We measured the out-of-order delay at the receiver MPTCP end as the difference in time between the data sequence number acknowledgments generated for the sender at the receiver.

Figure 5.16 shows our observations in all our experiments. We observe that MPTCP is able to reorder and deliver a data stream to the upper layers within 50ms all MPTCP data transfers in 90% of the time, regardless of the vehicle's velocity and the data transfer size. This result is explained by our previous observations in Section 5.4.2.1: there is no significant distinction between the RTT performance in both the cellular and the WiFi paths, thus, MPTCP at the receiver end does not struggles to reorder the arriving data segments. When a TCP subflow is added, that is, when the vehicle enters the coverage of a new WiFi AP, be it TWCWiFi or our WiFi RSU, the server starts delivering packets onto the new path once the connection has been established. During this period, the out-of-order delay of the transmission keeps within the 50ms, according to our observations because the path characteristics of the WiFi and the cellular paths are similar, as the RTT/Loss Tables 5.5 and 5.6, and Throughput Table 5.3 show. Once the vehicle moves away from the WiFi AP coverage, MPTCP shifts traffic back onto the main path, the cellular connection in our experiments. The transmission resumes and the data stream is reordered and delivered to the

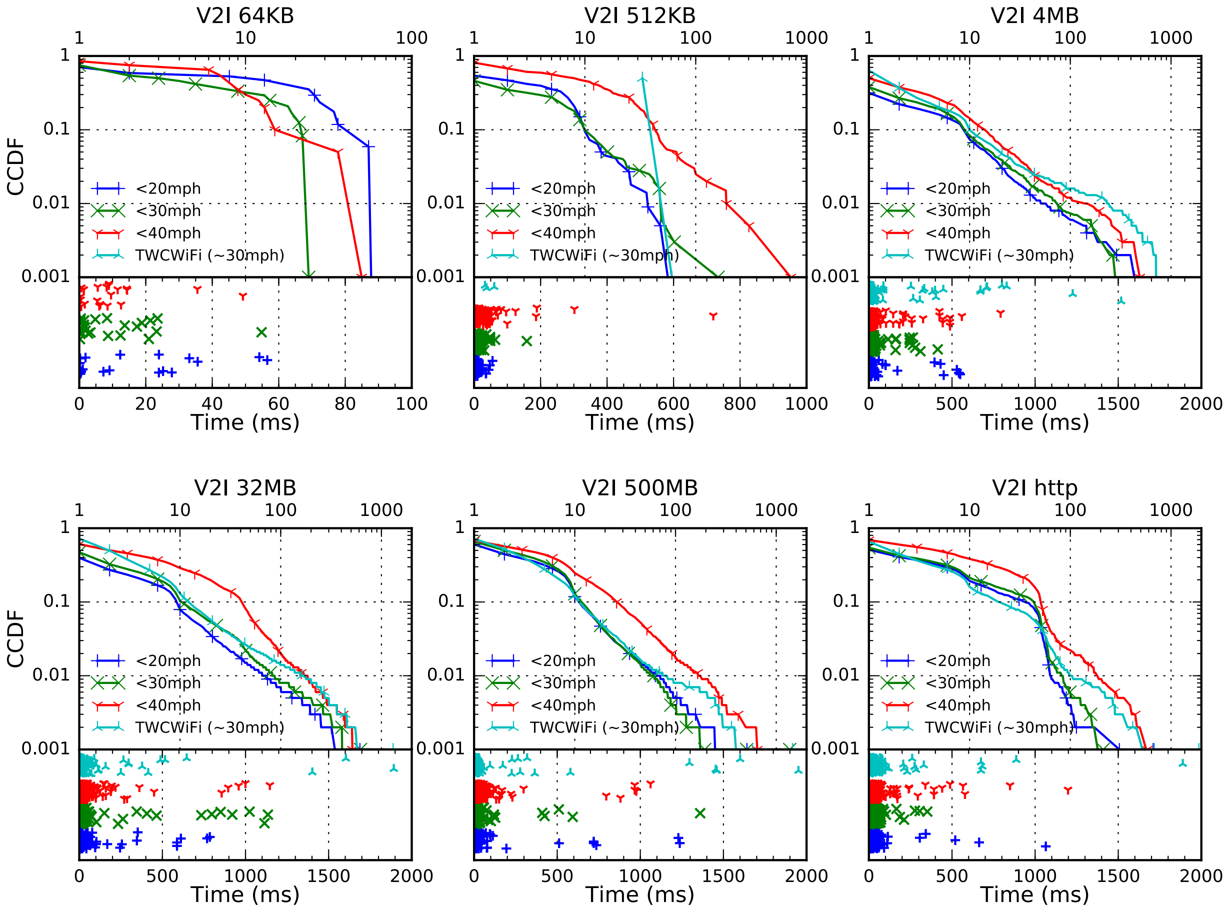


Figure 5.16: CCDF of the out-of-order delay for all MPTCP sessions

upper layers at the receiving end.

5.5 Discussion of the Results

In this section we pull together our findings regarding the use of MPTCP on highly dynamic VANET scenarios. Considering the current advancement in wireless networks, particularly on cellular carriers, as well as the deployment of WiFi networks in major urban areas, the availability of multiple paths to VANETs is a reality. VANETs benefit of from these new network resources to realize their potential benefits described in Section 2.1.

Regarding MPTCP, we found that in general the vehicle's velocity does not pose a significant impact on the download time performance in any of our file sizes used. The maximum velocity used in our experiments is 40mph, or about 64km/h, so we do not draw any conclusion for velocities above that; in fact, we attempted higher velocities but we found it to be pointless due to the short connection time. In terms of download time and throughput, MPTCP has similar or slightly worse performance due to TCP subflow disconnection recovery. On the other hand, the major advantage of MPTCP can be observed in terms of traffic offloading. MPTCP is able to offload the cellular path by as much as 60%; this is a major result that may help reduce the dependence of cellular carriers for Internet service and reduce connectivity costs. In terms of path discovery, we found that MPTCP is able to establish a new TCP subflow within one data RTT delay once the WiFi path has been discovered; RTT for data packets are considerably larger in size, around two orders of magnitude larger compared to control MPTCP packets, therefore, there is enough time to establish a connection within the time a data packet is delivered. This means that the time to start transferring data on a newly discovered path is negligible; the problem, however, consists in discovering that path in the first place. We found that our current equipment takes a few seconds to detect the AP and obtain an IP; this is a significant time that could be solved by specialized hardware, for example, we used a high gain antennae. Nevertheless, this issue is out of the scope of the MPTCP protocol. Another interesting finding is how MPTCP behaves to TWCWiFi AP diversity. When the vehicle stops at intersections or when there are several contiguous AP on the road, MPTCP is able to leverage them to obtain high traffic offloading.

In terms of latency, in general we found that MPTCP performance is adversely affected, mostly due to the high mobility dynamics rather than the protocol shortcomings. Nevertheless, we found that MPTCP is not suitable for tight latency constraint applications. If the latency constraints are relaxed, as in the case of non-live video transfers or large file transfers, MPTCP may provide performance benefits, mostly in terms of traffic offloading. We expect that video transfers that use buffering would suffer hardly any impact. Another latency aspect considered in MPTCP transmissions is the out-of-order delay. We found that this metric is highly dependent on the path characteristics of the TCP subflows in use. In our experiments, we found similar characteristics in both our cellular and WiFi paths; therefore, this delay had no significant impact in our findings, in general.

Lastly, another major finding is the vulnerability of MPTCP long file transfers due to losses and latency effects. We observed that the longer MPTCP connections are exposed to the vehicle traffic dynamics, the worse the performance it has, especially in terms of download times. For example, while large and very large file transfers hop among distinct APs, short transfers and HTTP flows open and close connections quickly; even if we transmit a large amount of data worth of short transmissions. We concluded that long transfers are exposed to mobility and traffic conditions on the road longer than short flows.

5.6 Related Work

This paper explores the feasibility and performance of MPTCP over a VANET under highly dynamic environments. Most of our presentation and analysis is based on the work of Chen, et. al., in [20]; we applied their study methodology of MPTCP over VANET, and we feel our work is complementary to their static study of MPTCP. Lee, et. al, also study MPTCP under mobility in [81]. They propose to integrate a mobility context state, i.e., walking, driving, onto the path selection; our study uses the MPTCP protocol as proposed by the standard [43]. Riaciu, et. al., studies the performance of MPTCP under mobility in [107]; however, their work only focuses on indoor environment under controlled scenarios while we focus on outdoor VANET scenarios. Paasch, et. al., studies the MPTCP performance over mobile/WiFi handovers in [101]; but their work focuses on controlled mobility and energy concepts. Chirwa, et. al., applies MPTCP to a platoon of cooperating drones in [22]; this work studies a scenario very close to the VANET but it is under simulation. Deng, et. al., measures the MPTCP performance over multiple interfaces in [30], but there is no mobility involved. Ferlin, et. al., takes a look at the role of path selection and scheduling in [41] over heterogeneous wireless networks; Nguyen, et. al., evaluate MPTCP in terms of throughput and load sharing in [98]; Zhou, et. al., studies cooperative MPTCP over LTE interfaces in [131], just like our V2V test bed; in all these works, there is no mobility involved. Han, et. al., studies the impact of MPTCP over mobile web applications in [52]; Pollalis, el. al., and Hampel, et. al., propose in [104, 51] a proxy server to allow non-MPTCP speakers engage into multi-path connections; these works are applications that work on top of MPTCP and are relevant to our architecture. Arzani, et. al., studies path characteristics and scheduling policies to evaluate

MPTCP performance in [7]; Paasch, et. al., gives an insight on MPTCP schedules in [102]; Jiang, et. al., studies the concept of bufferbloat in cellular connections in [68], Ferlin-Oliveira, et. al., and Chen, et. al., do it over MPTCP [42, 21]; Li, et. al., takes a look into the role of receive buffers in [84]; Baidya, et. al., goes one step further into proposing an algorithm to improve the issue of small receive buffer degradation of MPTCP in [8]; these last works explore fundamental concepts that impacted our MPTCP performance analysis under VANET scenarios.

5.7 Relevance to the Thesis

Vehicle-to-Infrastructure configuration in VANETs allows vehicles to leverage a WiFi metropolitan infrastructure for the service betterment of platoons; in our experiments, we exploited the Time Warner Company® metropolitan WiFi network in the city of Los Angeles, California. This thesis explores WLAN offloading as a mechanism to provide service continuity to applications running on a vehicular network. Multipath TCP is the chosen protocol that allows a VANET to hop through networks. It is well-known that TCP, the current state-of-the-art, is a break-before-make type protocol that interrupts service continuity upon movement or network disconnection; this makes TCP unsuitable for highly dynamic networks such the ones in VANETs. When a network path is lossy or disappears due vehicle movement, this work has shown that MPTCP is a viable candidate to maintain connectivity; we have shown that it works remarkably well at velocities up to 40mph, 65km/h. In terms of offloading, our experiments showed that MPTCP is able to offload traffic onto a nearby WiFi access point or RSU unit at rates up to 60% of the overall throughput. These results support the use of MPTCP on VANETs in order to provide service continuity to a platoon of vehicles solving vehicular tasks; this finding supports this thesis goal for service continuity in mobile devices.

CHAPTER 6

Vehicle-to-Vehicle Experiments

6.1 Introduction

The deployment of WiFi and cellular networks provide multiple opportunities for mobile devices to connect to the Internet. While both networks currently provide similar performance in terms of bandwidth and latency characteristics, they are distinct in terms of coverage and cost. The mobile user may favor the use of cellular networks due to its urban-wide coverage but the network carriers' pricing schemes could make it prohibitively expensive; therefore, offloading traffic onto freely available WiFi networks is desirable.

One type of mobile user is a vehicle on the road. Vehicles may use network connectivity to obtain information that could improve the safety of passengers, the vehicle, or the infrastructure, improve vehicular flow and mitigate congestion, decrease the pollution impact, deliver content for entertainment, etc. Unlike the mobile pedestrian, vehicles have highly dynamic movement and speeds, rendering the WiFi infrastructure useless at high velocities. However, vehicles could share their cellular connectivity if they form VANET platoons using WiFi LANs in V2V configurations. The question is how to maximize the use these multiple paths.

The current state-of-the-art in establishing an end-to-end connection from a moving vehicle through the Internet is via a cellular network. However, if vehicles establish TCP connections to share their resources but separate, the connection will inevitably break, making TCP not a suitable candidate. Multipath TCP, on the other hand, allows the establishment of end-to-end communication using multiple paths; it allows seamless addition and removal of paths as vehicles enter and leave their proximity.

In this chapter we study the performance of MPTCP on vehicular networks, VANET, under

V2V configuration. The main goal of this study is to understand if MPTCP is viable for V2V scenarios, and if it is, how well does it perform compared to the state-of-the-art. For this purpose, we build a real-life test bed and drove around the city of Los Angeles and the freeway system engaging on MPTCP transmissions. Our experiments are characterized by distinct vehicular velocities: 25mph, 45mph, and 65mph, as well as the type of traffic they carry: short-lived, long-lived, and HTTP traffic. The main findings are as follows:

1. MPTCP is viable on V2V VANET configurations;
2. velocity slightly impacts the MPTCP bandwidth performance;
3. velocity significantly impacts the latency, making MPTCP unsuitable for applications with tight latency constraints in V2V scenarios;
4. traffic offloading and seamless handovers are the most significant MPTCP advantages.

6.2 Vehicular Test Bed

We setup our test bed under Vehicle-to-Vehicle, V2V, configuration in order to test the performance of MPTCP under a VANET as shown in Figure 6.1.

6.2.1 Hardware

The hardware used on this configuration consists of a mobile *MPTCP Client* that connects to a *MPTCP Server* using two paths: first, via its own cellular interface; and second, via a *Relay Vehicle* that carries the same equipment configuration as the mobile client. The server is a commodity Intel PC running Ubuntu 14.04 LTS and MPTCP v0.90, equipped with a 1Gbps Ethernet card connected to an ISP fiber-optic service provided by Verizon FIOS® (now Frontier Communications). The server runs a web server, Apache, on port 5280 to separate our measurements from normal traffic; the traffic generated follows the description given in Section 6.3.2. The mobile client is a Lenovo Thinkpad® W520 laptop that also runs Ubuntu 14.04 LTS and MPTCP v0.90. It connects to the test bed via two interfaces: cellular and WiFi. The cellular connection is provided by an

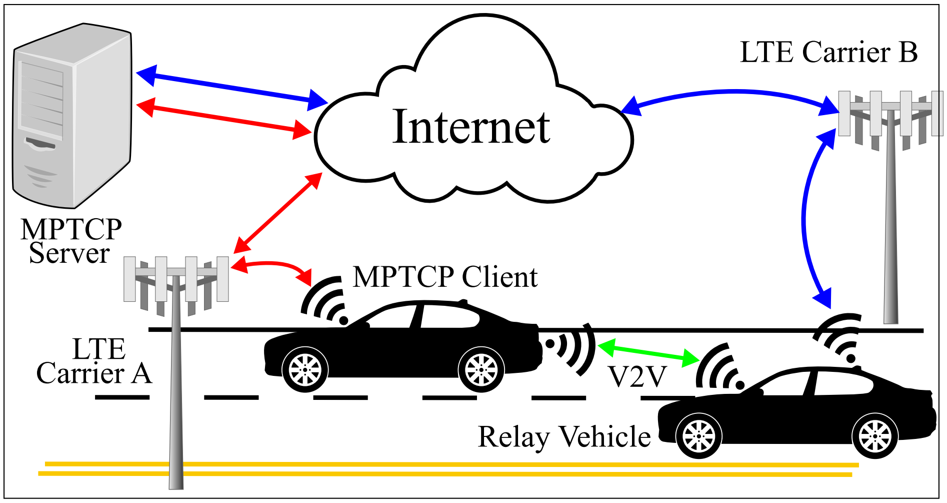


Figure 6.1: A VANET V2V configuration.

Apple iPhone® 5S via a USB cable; the phone establishes a LAN with the laptop through the USB cable in order to eliminate any wireless interference. The relay vehicle shares its cellular interface with connected clients using a WiFi LAN; at the moment of the measurements, both the relay and the client vehicles are under motion forming a VANET platoon. The WiFi interfaces on the vehicles use the Alpha AC1200® USB dongle that provides connectivity on the 5GHz IIS band. The reason for this selection is to emulate the DSRC communication (5.8GHz over 10MHz bandwidth channels) common in VANET scenarios with undergoing car-to-car communication. The V2V network between the vehicles is an IEEE 802.11n WLAN operating on channel 161 (5801MHz, 20MHz bandwidth). The client connects to the server via distinct cellular providers. The Relay Vehicle does not implement MPTCP, however; its only function is to relay traffic from its clients to the Internet via its cellular interface, thus behaving like a middle box. In our experiments, unless otherwise stated, *Carrier A* is AT&T LTE and *Carrier B* is T-Mobile.

6.3 Methodology

Our experiments consists on repeated downloads of a particular fixed-sized file for twenty times, in order to avoid isolated network instabilities; we also generate HTTP traffic based on exponentially distributed HTTP requests described in Section 6.3.2 to emulate real users. We perform the experiments at distinct times, i.e., morning, noon, evening, etc., in order to mitigate the impact of network load variations. In some scenarios, however, it is not possible to perform experiments at distinct times due to the vehicular congestion in Los Angeles, particularly on the high speed experiments. We run the experiments that required high velocities during late night hours or very early weekend morning hours. Table 6.1 shows the files used in our experiments and Section 6.3.2 describes how the traffic is generated.

To capture the traffic, we used the *tcpdump* tool at the MPTCP source and destination end points to generate the transmission *pcap* file. We then split the traffic for each TCP subflow using the *tshark* and *tcptrace* utilities. For each flow, we extract the correct RTT (on the server side) and loss during the transmission. We use the pcap file on the client side to get the throughput, the traffic offload for each path, as well as the out-of-order delay. We use the *curl* utility's output statistics to obtain the download times as observed by the application layer (goodput), while the *ifstats* gives us the throughput at the interface level.

During our traffic generation, our tools keep constantly sending *ping* packets on both interfaces in order to avoid connection setup delays unrelated to the MPTCP protocol; for example, RRC base stations on the cellular path may forget our cellular connection under long periods of inactivity, or the WiFi hardware may disconnect due to power saving features.

6.3.1 V2V Scenario Description

In our V2V use case, we used at most two vehicles equipped with the hardware described in Section 6.2.1. One vehicle, the *MPTCP Client*, engages into MPTCP connections with another static end that we call *MPTCP Server* while operating at several velocities on routes described in Section 6.3.3. Each client may use its cellular interface to connect to the Internet to generate network traffic as discussed in Section 6.3.2. It may also use the cellular connection from a second vehicle, the *Relay Vehicle*. This vehicle does not necessarily have MPTCP capabilities, its purpose is simply to form a small VANET platoon with the MPTCP client and share its connection. Our experiments are characterized by the velocities in which operate on selected routes with intrinsic operating characteristics further discussed in Section 6.3.3.

Type	**Repeated 20x**	**HTTP Traffic**
Small	64KB, 512KB	[1,99]KB, 1KB incr [100,900]KB, 100KB incr
Large	4MB, 32MB, 500MB	100MB, 200MB, 500MB

Table 6.1: Distinct file sizes used in our experiments.

6.3.2 Traffic Generation

6.3.2.1 Short Flows

Twenty downloads of 64KB and 512KB files make short-lived flows. We are interested in understanding the performance of these flows under highly dynamic scenarios as most of the web traffic on the Internet is short-lived. Also, short-lived flows are particularly wasteful of network resources due to the high ratio between overhead and overall data delivered [115].

6.3.2.2 Long Flows

Twenty downloads of 4MB and 32MB, one of 500MB make long-lived flows. We are interested in observing the performance of MPTCP over long, sustained transmissions while exposed to path disconnections under highly mobility VANET scenarios. Video transfers, video conferencing, or large vehicular data files are examples of these flows.

6.3.2.3 HTTP Traffic

It is possible that repeated downloads of fixed-sized files may not accurately represent the overall traffic dynamics for the common user; for this reason we created scripts that closely represent this behavior. Our HTTP traffic model generates data bursts that represent a web page download and multiple bursts represent web traffic. Small fixed-sized files from Table 6.1 or large [1MB, 4MB] in 1MB increment files make each burst. Small files have a configurable probability ρ to be selected, while large files have a probability of $1 - \rho$. In our experiments a probability of $\rho = 0.80$ gives us a good balance between small and large files. The overall burst size is randomly selected with a maximum size of 10MB. Each burst transmission is separated by a small time delay that follows a Poisson distribution with $\lambda = 10ms$; this value produces a continuous and sustained data transmission without behaving like CBR traffic. The overall data transmitted in all the data bursts is, unless otherwise noted, 100MB worth of downloaded data. There are situations where we use 300MB or 500MB because the experiments turned out to be too short, i.e., freeway experiments, see Section 6.3.3.

6.3.3 Routes

The routes used in our test bed are in the city of Los Angeles, California, matching the 25mph, 45mph, and 65mph velocities to the best of our abilities and with the lowest possible risk. Each route selected is one that respects the city posted speed limits. Moreover, the driver never operates the equipment nor attends to it while the vehicle is under operation. We note a few curious police officers regarding our work, mostly interested about our external antennae; some even shared technical aspects of their own propitiatory systems. No citation nor traffic infraction was imposed at any time to any of the participants of this work.

Figure 6.2 shows the routes for the 25mph and 45mph routes. For the 25mph route shown in blue in the figure, the vehicles drive close to each other clockwise around a loop of about 1.2km. The streets in the neighborhood have a 25mph posted speed limit and the terrain includes hills on the north-west corner. As the experiments progress, foreign vehicles to out test bed pass by or merge between our vehicles (due to traffic stops). The route includes multiple vehicles street-parked, trees, pedestrians walking their pets, and other obstacles that would typically be found in a neighborhood.

The 45mph route is shown in red color in Figure 6.2 as well. The route is a 3.2km straight path on Venice Bl. (CA 187) that starts at Sawtelle Bl. and runs through Lincoln Bl. (CA 1, PCH) until the vehicles may perform a safe U-turn. The posted speed limit is 45mph at the time of the experiments; although, it has currently changed. The vehicles engage into multiple intersection, stop/light signs and crossing traffic; even though our vehicles may stop, we perform our experiments during traffic ours that could accommodate fluid traffic.

Lastly, Figure 6.3 shows the 65mph route in blue color. This route is about 56km long and

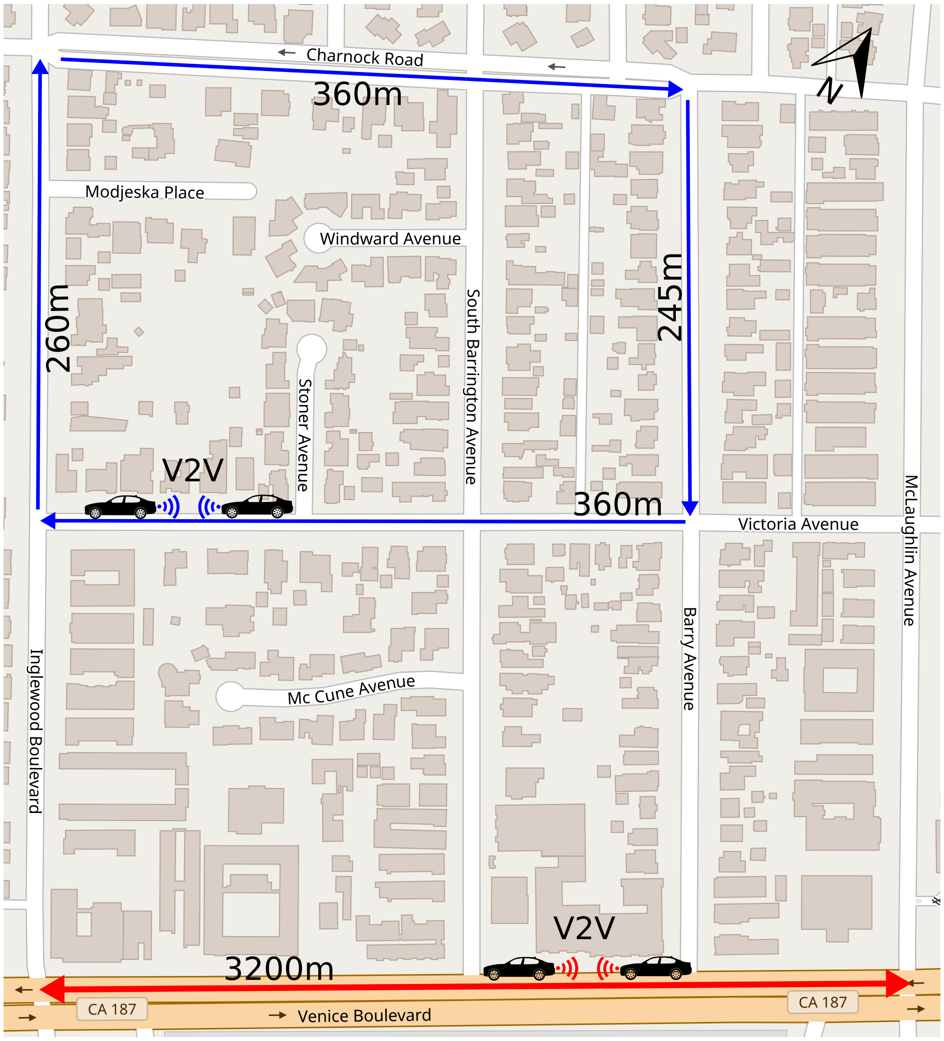


Figure 6.2: 25mph and 45mph routes used in Los Angeles

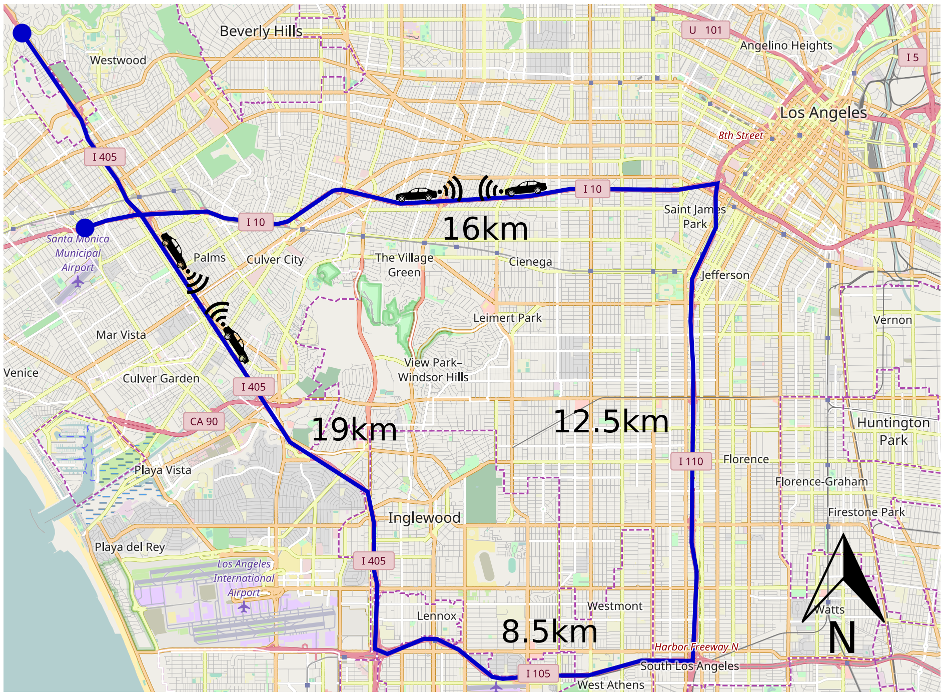


Figure 6.3: 65mph route used in Southern California

our experiments are carried out during the weekends at very early or very late hours in order to avoid traffic jams. The route starts at Sunset Bl. entrance to the I-405 freeway southbound; the vehicles proceed for about 19km until they reach the I-105 just after the LAX airport. On I-105, the vehicles run for around 8.5km eastbound until they merge onto the I-110. On the I-110, they drive northbound for another 12.5km before they reach Downtown LA. In Los Angeles, the vehicles enter the I-10 and drive for another 16km westbound until they exit on Bundy Dr. just before Santa Monica, CA.

6.3.4 Metrics

The experiments on this paper used the following performance metrics.

Download Time. Time spent between the SYN packet of an HTTP request to the last data packet received in that flow. This metrics shows the performance perception at the application layer.

Throughput. Bytes delivered by the interface per unit of time, including lower-layer overhead. This metric shows the actual data transmitted per TCP subflow.

Traffic Offload. Ratio between the data transmitted on the main path and the WiFi path. This metric measures how well MPTCP balances traffic among TCP subflows.

RTT. The round-trip time between source and destination. This metric impacts the MPTCP scheduler's TCP subflow selection when delivering the next packet.

Out-of-Order Delay. Time necessary to assemble the data stream from all TCP subflows before it is delivered to the application layer. This metric is particularly important for very asymmetric path characteristics, as in V2V scenarios.

Packet Loss. The ratio between retransmitted packets and total packets sent in a connection. This metric is fundamental to TCP subflow performance.

6.4 Experiments

In this section we present our findings regarding the performance of MPTCP connections between moving vehicles engaged on a VANET platoon under V2V configuration. The current state-of-the-art for vehicle Internet connectivity is via a cellular network in a single-path TCP, SP-TCP, connection. The MPTCP client uses its cellular connection as well as the relay vehicle's WiFi LAN in V2V mode; we refer to the latter as the WiFi path. Other network technologies like DSRC and WiMAX may be used "without loss of generality" as MPTCP is a transport protocol; however, the LTE network is currently the easiest radio technology available to us. Unless otherwise noted, we use AT&T for the observed vehicle, and T-Mobile for the relay vehicle.

6.4.1 Bandwidth Results

Bandwidth is a metric that measures how much data a connection is able to transfer per unit of time. The TCP/IP stack and the lower layers use protocols to provide network delivery services to the Application layer. These protocols use small headers prepended to each data packet; the accumulation packet headers is called *network overhead*. Bandwidth measurements may or may not include network overhead. The download time is a representation of the goodput in terms of time without network overhead; the throughput, in turn, includes the network overhead in its

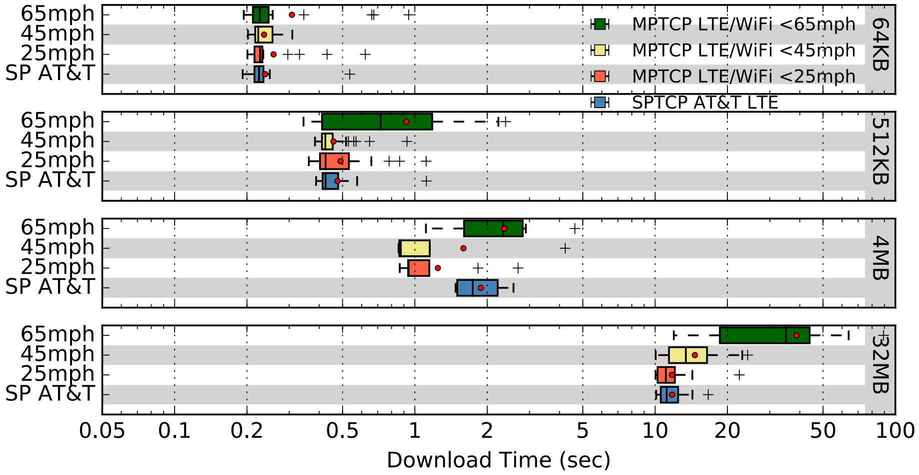


Figure 6.4: Download time for short and long-lived flows

Cellular Carriers

Carrier	RTT (ms)	Download (Mbps)	Upload (Mbps)
AT&T	34.80 ± 2.97	27.45 ± 1.50	3.86 ± 1.18
T-Mobile	49.30 ± 2.63	5.80 ± 1.55	1.10 ± 0.49
Verizon	37.20 ± 4.49	7.38 ± 2.41	1.17 ± 1.27

Table 6.2: Speedtest® measurements of LTE providers

measurements.

The download time measurements for short and long-lived connections are summarized in Figure 6.4. Figure 6.5 shows the download time for very long, continuous MPTCP connections and HTTP traffic. A visual representation of the throughput is shown in Figures 6.6 to 6.9; we point out particular issues that we found interesting to address in this figure as well. In general, these measurements gives us an insight on how efficient MPTCP is at the transmission of data under distinct vehicle velocities compared to SP-TCP connections via a cellular path.

6.4.1.1 Short-lived flows

Figure 6.4 shows the short-lived connections for file transfers of 64KB and 512KB in size. We first notice that the download time of MPTCP connection compared to SP-TCP using the cellular path is practically the same for 64KB flows; the same is for 512KB for flows under 45mph or less. This result shows that MPTCP short-lived connections are able to match the SP-TCP performance and at the same be fair with other TCP flows on the same path, in accordance to goals 2.1 and 2.2 in Section 2.2. Moreover, the added overhead of managing two TCP subflows in MPTCP compared to SP-TCP does not seem to affect the download time, considering that short-lived flows are known to be wasteful, see discussion in Section 6.3.2.1. 512KB flows at 65mph, however, show a performance degradation of about twice the time to download the file compared to the other MPTCP connection under slower speeds or the SP-TCP; also there is large variability in these measurements. This is a clear indication that large velocities have an impact on the MPTCP connection.

6.4.1.2 Long-lived flows

Figure 6.4 also shows long-lived connections represented by 4MB and 32MB file transfers. We first look at the 32MB results since they follow the same pattern that we saw in short-lived flows; they also represent the general behavior of MPTCP throughout our experiments. Note that at 25mph the download times are practically identical to SP-TCP; then there is a slight performance degradation at 45mph and noticeable underperformance at 65mph. In general, the MPTCP download time performance gradually degrades as the vehicle speed increases. This is expected because long-lived flows are exposed longer to the vehicular moving dynamics and the coupled congestion control algorithm LIA. Despite this behavior, MPTCP maintains its connectivity in almost all the cases (see Section 6.4.1.4 for cases when it does not) with traffic offloading present throughout the transmission (see Section 6.4.2). This result shows that velocities play a role at MPTCP performance in long-lived file transfers; however, this impact does not render MPTCP useless since it is able to, first, complete the transfer, and second, perform comparably to SP-TCP. This result demonstrates that MPTCP is able to operate under such highly dynamic scenarios successfully for long-lived transmissions.

4MB flows depict a similar behavior as 32MB transfers, but we encounter an interesting scenario that we found worth of discussion. We observed that MPTCP may, in some situations, outperform SP-TCP, which is not necessarily a desirable outcome, especially on the Internet where there are other competing TCP flows. If MPTCP beats the performance of SP-TCP, it means that

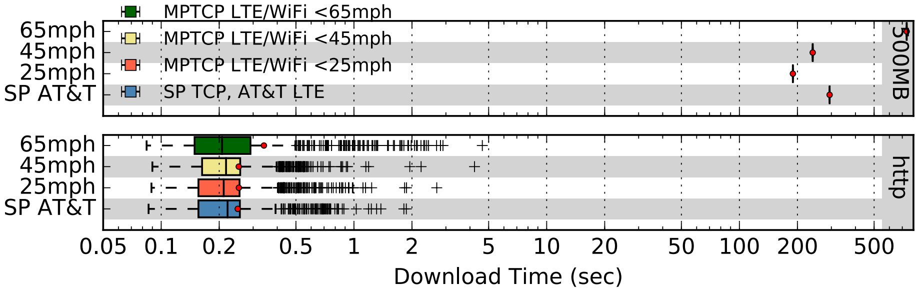


Figure 6.5: Download time for 500MB and HTTP traffic

the connection is not fair with other SP-TCP transfers; thus it fails at achieving its fairness goals. In this particular case, these 4MB flows outperform SP-TCP by almost twice in download time at 25mph and 45mph, compared with those at 65mph and SP-TCP. In Section 6.4.2 we show that in fact these transfers perform practically as SP-TCP transfers since little traffic offloading is present on the WiFi path. How does this happen? We immediately disregarded temporal factors since we averaged our measurements at distinct times of the day and the experiments did not record any disconnection on the WiFi path during our experiments that could justify this behavior. Table 6.2 shows the baseline download/upload time performance of the cellular providers used in out test bed in a non-mobile scenario. Note that AT&T download performance is $4.7x$ that of T-Mobile, the carrier used in the relay vehicle offering the WiFi path. Considering that the routes used in our test bed in Section 6.3.3 for the lower velocities (25mph and 45mph) are on the same, small geographical area, it is possible that there existed temporally, localized spatial network issues, i.e., T-Mobile signal weakness (coverage holes in LTE terminology). We repeated the same experiments on another day-hour time, at another location, and we could not replicate the results shown by these particular 4MB flows. We conclude that at the time of this particular experiment, the vehicle drove through a coverage hole on the route followed by our vehicles. What is important to note, however, is that MPTCP correctly allocates most of the traffic onto the cellular path in this highly asymmetric path characteristics scenario in a moving situation. MPTCP behaves correctly, like a SP-TCP connection with little to no offloading.

6.4.1.3 Continuous flows and HTTP traffic

Figure 6.5 shows the download times for two kinds of network traffic: continuous flows (500MB) and HTTP traffic (100MB). As described in Section 6.3.2, continuous flows may represent large, sustained traffic downloads with a single HTTP requests while the HTTP traffic contains many HTTP requests. We start with the 500MB flows. We observe again the same behavior as in short and long-lived file transfers; that is, the vehicle's velocity has a gradual degradation impact on the performance as it increases. Note that the download time at 25mph and 45mph have similar performance to SP-TCP transfers but the performance at 65mph does have a larger impact, roughly about $2.4x$ that in SP-TCP. In general, though, these results show that MPTCP transfers are also viable for very long continuous transfers.

For HTTP traffic, there is no significant difference between the first two velocities and SP-TCP; however, there is a minor impact for transmissions at 65mph, consistent with our previous observations. As explained in Section 6.3.2, HTTP traffic represent multiple HTTP requests of short files that resemble web objects; thus, the behavior should be similar to those in short-lived flows. Note, however, that there are two clusters in the whisker plot: the data described by the box (second and third quartile) and the outliers; the former represents the short files while the latter represents the large files of size 1MB to 4MB, see Section 6.3.2. In the worst case, MPTCP takes about five seconds to complete a download of any web object at any speed. Short files take in average $200ms$ for download at or below 45mps, $300ms$ for 64mph. Large files are downloaded within a second for velocities up to 45mph in average; more spread results are observed at 65mph. These values are similar those observed by SP-TCP, however, especially for velocities under 45mph. This result shows that MPTCP performs well on HTTP traffic that includes multiple HTTP requests under distinct dynamic conditions. We will see in the next section, however, interest issues that arised in HTTP traffic scenarios.

6.4.1.4 Throughput

Figures 6.6 to 6.9 shows the visual representation of the per-TCP subflow throughput observed at the mobile vehicle generated by the MPTCP traffic. The figures represents the MPTCP traffic

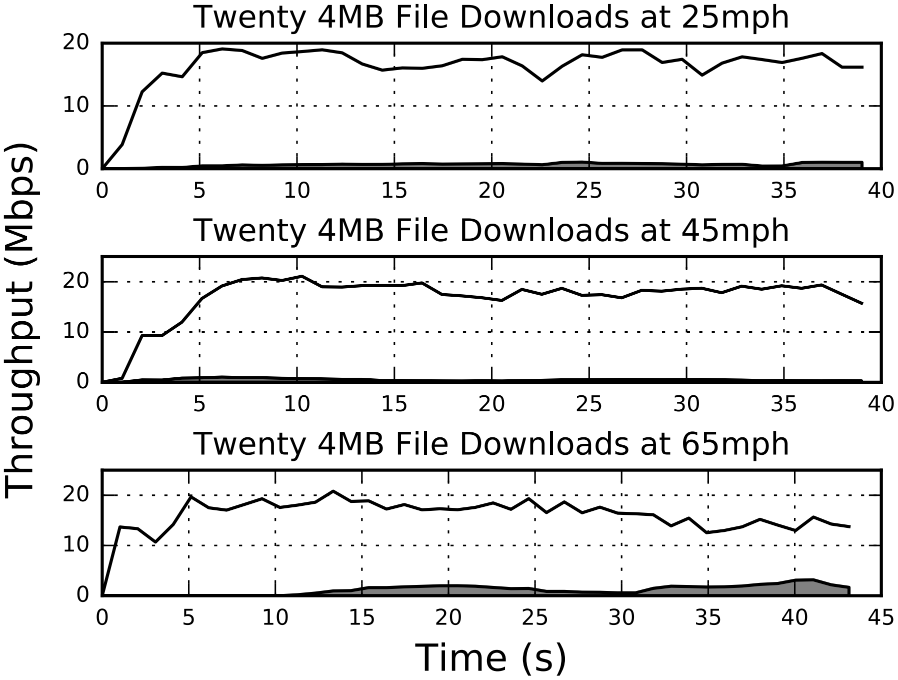


Figure 6.6: Throughput observed for 4MB flows at varying speeds

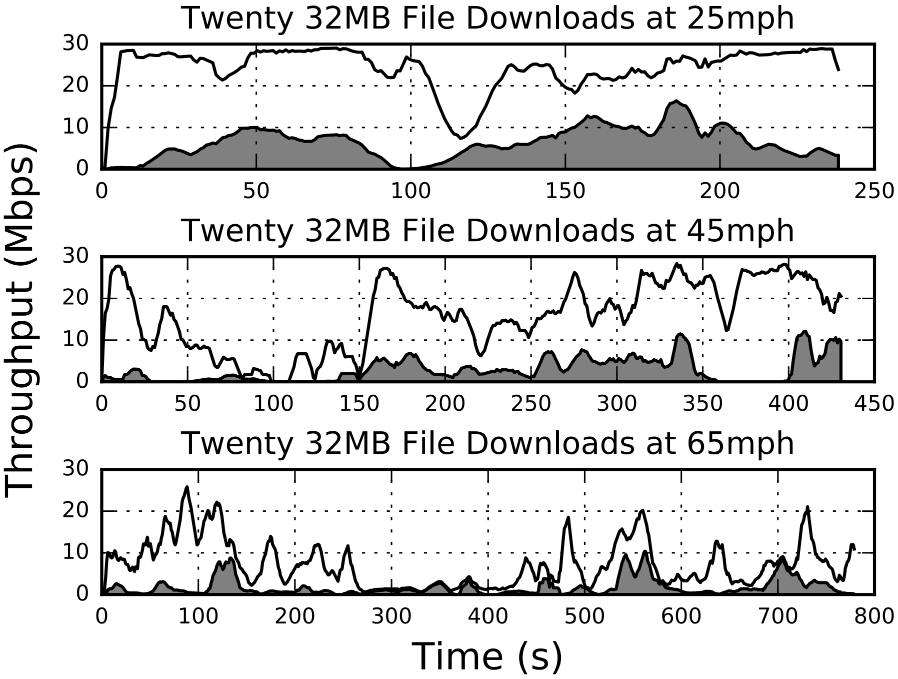


Figure 6.7: Throughput observed for 32MB flows at varying speeds

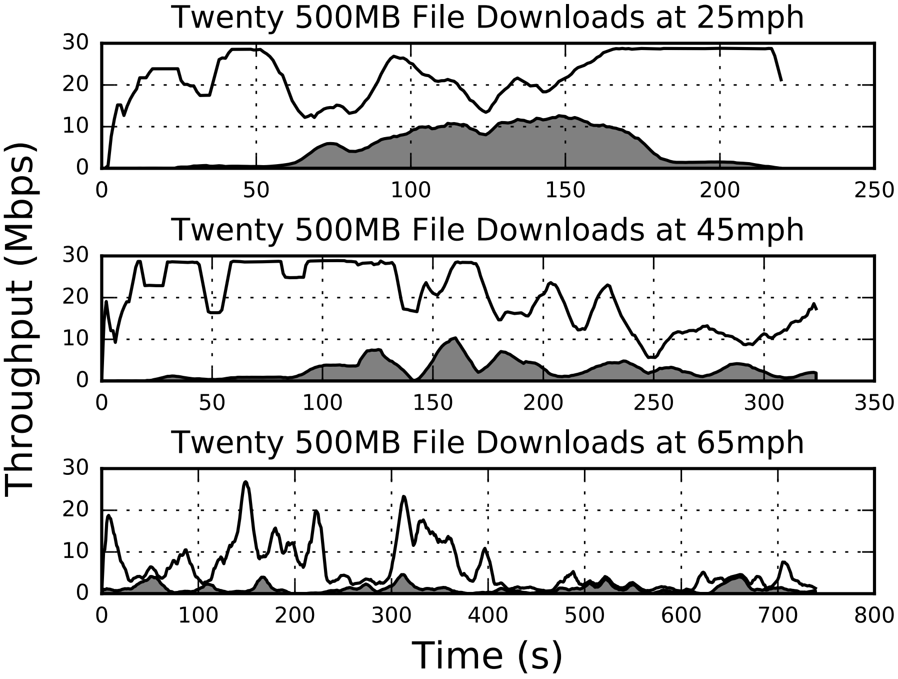


Figure 6.8: Throughput observed for 500MB flows at varying speeds

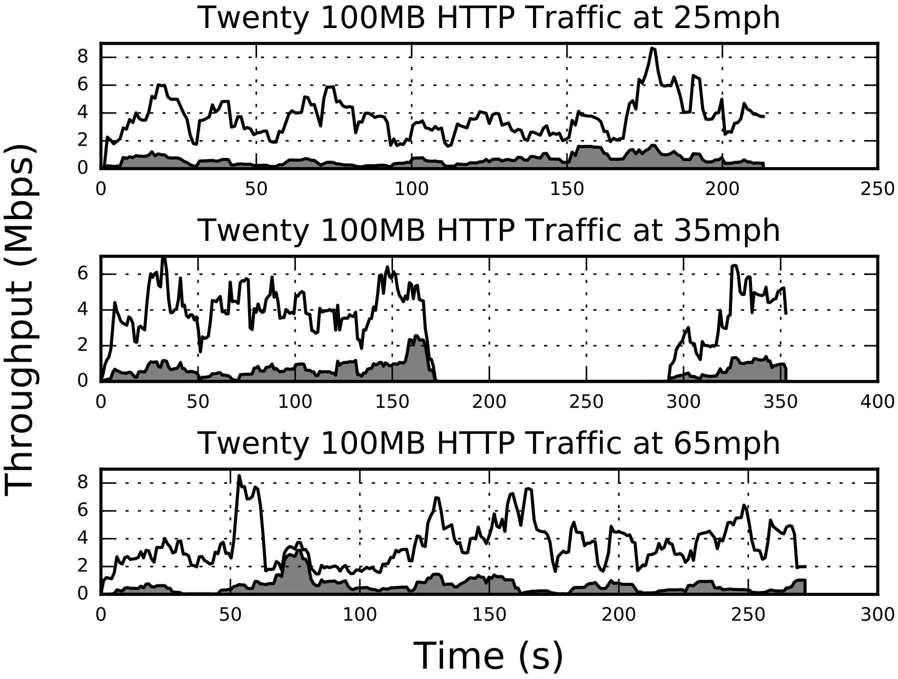


Figure 6.9: Throughput observed for HTTP flows at varying speeds

for 4MB, 32MB, 500MB and HTTP flows, as described in Section 6.3.2; we do not include the 64KB and 512KB subfigures due to space concerns. Each figure has three plots that represent each velocity used in our experiments. Each plot shows the *throughput* observed per unit of time (in seconds) during the entire transmission. The shaded area in each plot shows the network traffic offloaded onto the WiFi path and the line plot represents the cumulative throughput observed during the entire experiment; that is, the addition of cellular and WiFi path's bandwidth.

Figure 6.6 shows the issues described in Section 6.4.1.2. Notice that there is little to no traffic moved onto the WiFi path for velocities 25mph and 45mph, and some traffic at 65mph. Since the WiFi path includes the T-Mobile link, we can see the coverage hole affecting the relay vehicle; it is not a WiFi failure (V2V link) since the path never disconnects as offloading is continuously observed, see Section 6.4.2. This shows that MPTCP responds well to these kind of challenging scenarios while matching or, as in this case, beating SP-TCP.

In Figure 6.7, 32MB flows show how MPTCP normally behaves throughout the rest of our experiments. The most important observation from all figures is that MPTCP is able to offload traffic at any vehicular velocity. We also note that the throughput from each path is correlated as expected since the connection uses LIA, the coupled congestion control algorithm. The impact of velocity is clear as well; slower velocities lend themselves to larger offloading performances but note that even at the fastest velocity of 65mps we still see consistent throughput at the WiFi path. For the long, continuous flows, 500MB, we see the performance at 25mph is much better than the one at 45mph and 65mph; see Figure 6.8. In the 25mph experiment, we see a healthy participation of the WiFi path on the period between the $50s$ and $200s$ marks; in fact, the WiFi path carried most of the load at this period. The other two experiments using the larger speeds show gradual decrease in the use of the WiFi path; however, at 65mph we can see a larger bandwidth degradation compared to the other two experiments. This shows again the impact of high velocities on the MPTCP performance. 32MB and 500MB transfers are long-lived flows that are exposed longer to the network traffic conditions and vehicular movement dynamics. Despite these conditions, our results show that in general MPTCP is capable of maintaining an ongoing connection with its peer relay vehicle over the WiFi path for as long as the connection lasts or the peer vehicle is present. Finally we show the time series plots for the HTTP traffic scenarios at distinct velocities in Figure 6.9. Since most of the HTTP traffic is composed of small files, they represent short-lived transmissions (64KB and 512KB look the same). Note that there is some, continuous offloading throughout the 100MB worth of HTTP traffic with no significant impact by speed as in the short-lived flows (see our conclusions in Section 6.4.1.1).

We found another interesting issue happening in our HTTP experiments, shown at the 45mph velocity plot: the bandwidth flattened during the period between $163s$ and $298s$ marks. First we thought that a failure on the WiFi path could have brought down the MTPCP connection, but after reviewing the pcap files we found that the real culprit was a disconnection from the mobile vehicle's cellular path on AT&T, the MPTCP connection's main path. MPTCP has a main path and a set of additional paths added after the connection is established. If the additional paths are active and connected, they all contribute to the overall MPTCP bandwidth despite the loss of the main path. Technically, if the main path fails, another interface in the mobile device takes over as the main path, provided that the operating system detects the network disconnection. In our case, we had particular situation: the cellular link provided by AT&T on the iPhone device disconnected, not on the link that connects the iPhone to the laptop that carried out the MPTCP connection; therefore, laptop never detected the network failure. This caused our MPTCP socket to timeout on the SYN packet when a new web object was requested. The socket timed out and tried again, three times for MPTCP and three additional times for TCP upon protocol downgrade [43]; that made the $135s$ gap. Unfortunately, the SYN retransmission attempts happened all on the cellular path that failed! Why? Because it never stop being the main path. Why would it change? The first hop link never failed, a link in the middle of the network did, which it is out of the MPTCP scope (end-to-end protocol). We see this, however, as a issue in MPTCP because it is not able to use its other interfaces to establish the connection. This event demonstrates that MPTCP is unable to take advantage of its multiple interfaces when the main path fails in the middle of the network and before the connection discovers new paths. This issue may send the MPTCP connection to long stalling periods that are undesirable when in fact another interface may connect successfully. This issue is not subject to the goals of this paper, however, it is an interesting finding that requires further research; we leave that as a future work.

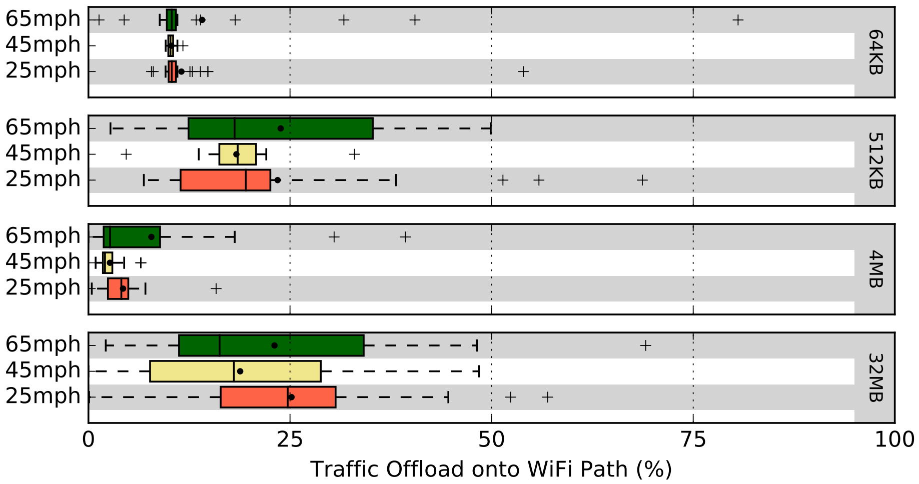


Figure 6.10: Traffic offloading on short and long-lived flows

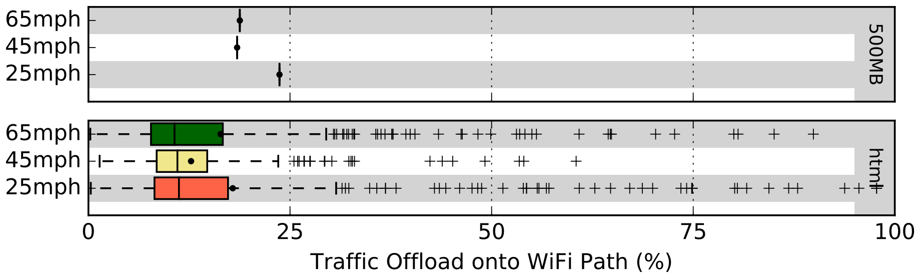


Figure 6.11: Traffic offloading on 500MB and HTTP traffic

6.4.2 Traffic Offloading Results

In Section 6.4.1 we saw that having similar bandwidth to SP-TCP flows is desirable because otherwise MPTCP would fail to meet its fairness goal as its designers envisioned it. Unless MPTCP finds a path that is completely unused, MTPCP should never "take more bandwidth than any competing TCP flow in a path" (goal 2.2). Therefore it is natural to ask: what is the advantage of using MPTCP over TCP? In the context of wireless networks, seamless handovers and traffic offloading. Seamless handovers maintain the end-to-end connection in MPTCP while this is impossible in TCP. Traffic offloading is desirable to relieve the cellular path; especially if the cellular carrier charges based on the traffic load (per kilobyte used) or if used during rush hours or during temporal down times (coverage holes).

Figure 6.10 and Figure 6.11 show the traffic offloading in all our experiments; as discussed in

Section 6.4.1.4, Figures 6.6 to 6.9 help us see the throughput of each interface visually. We only show the traffic offloading on MPTCP connections, as SP-TCP flows do not offload traffic. For short-lived flows, the 64KB transfers record an average of 11% while 512KB flows can offload a larger amount, 22%. For short-lived transfers, the vehicle's velocity does not seem to have any significant impact on the transmission. In fact, as we discussed in Section 6.4.1.2, the reason that the offloading recorded in all experiments is largely due to the WiFi path's cellular link characteristics shown in Table 6.2, and not the V2V WiFi link uniquely. For long-lived transfers, we see the offloading for 4MB flows onto the WiFi path at barely over 5%. As discussed in Section 6.4.1.2, this experiment is a special case and does confirm that the whole transmission behaved like a SP-TCP in practice. For long-lived flows we have the following results: for 32MB flows, we recorded an offload of about 23%, in 500MB flows we got a little over 20% in Figure 6.11. Finally, for HTTP traffic we saw 15%. These results show that MPTCP is able to offload a significant fraction of the load onto a peer's cellular connection regardless of the velocity that both vehicles operate on, as long as the relative velocity between the two vehicles is close to zero, that is, they move on the same direction at relatively the same speed. It is also clear that without seamless handovers, no traffic offloading can be achieved; this results show the main advantage of MPTCP over TCP in wireless scenarios.

6.4.3 Latency Results

Latency is an important metric that characterizes the system responsiveness. While the bandwidth gives us an indication on how much data can be sent through a path, the latency tells us how quickly we get a response from the other end. This is a desirable characteristic for delay sensitive applications such as voice-over-IP or live video conferencing. In this section we explore the latency performance of MPTCP on V2V scenarios with two metrics: out-of-order delay and round-trip time.

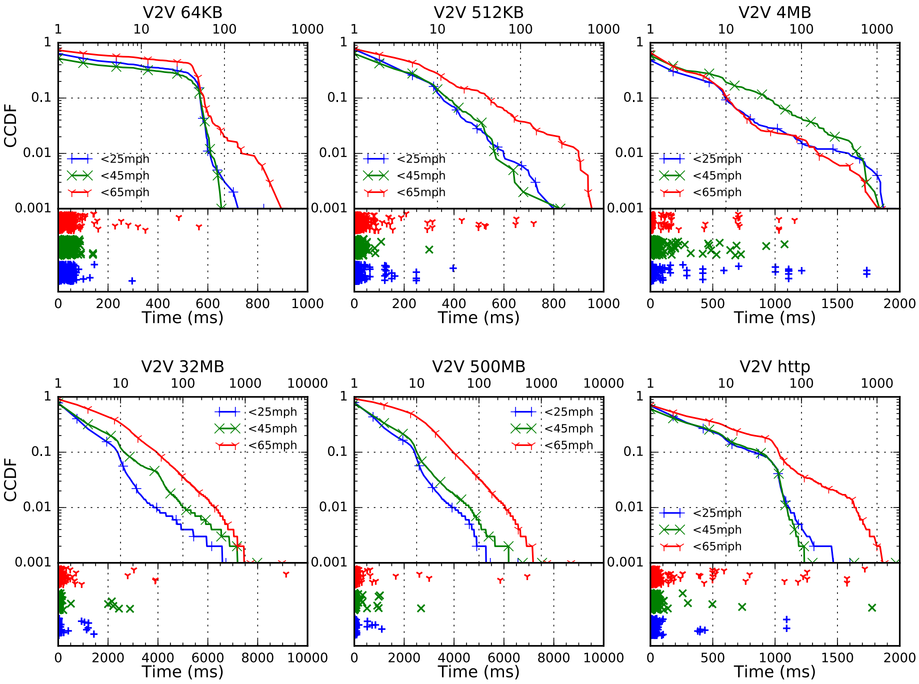


Figure 6.12: CCDF of out-of-order delay in MPTCP flows

6.4.3.1 Out-of-order delay

Figure 6.12 shows the complementary cumulative distribution of the out-of-order delay at the receiver end in all our experiments. The figure contains six subfigures showing the small and long-lived flows, the continuous flow and the HTTP traffic delays; in turn, each subfigure plots the delays per vehicle velocity experimented. Additionally, we provide a pseudo-spectrogram plot that shows where in the time axis the delay concentrates the most.

For the short-lived flows, the experiments show out-of-order delay values below the 100ms 95% of the time, regardless of the velocity used. However, the 65mph velocity has the worst performing delay recorded but it never takes more than $1s$ to reorder all the packets. Long-lived flows have the same probability to stay below the $100ms$ mark for reordering as well. Note the 4MB file transfer, a special case discussed in Section 6.4.1.1. The subfigure shows that there is practically no difference in the velocity used since most of the transmission is done through the cellular path (only about 5% offload, see Section 6.4.2); on other words, there is little to reorder. Note also the spectrograph plot, the outlier values appear without any connection with the vehicle's velocity for 4MB transfers. This result shows that the 4MB experiment clearly behaved as a SP-TCP.

The 32MB and 500MB subfigures show again the typical behavior in most of our MPTCP experiments. In both cases there is a 95% probability to stay below the $100ms$ in reordering packets and the 65mph velocity is again the worst performing scenario. Another aspect to notice is that short-lived flows have a maximum out-of-order delay of $1s$, while long-lived flows could suffer up to $9s$, as shown in the 500MB transfers. This is expected as long-lived flows are exposed longer to the vehicle moving dynamics. It also indicates that MPTCP requires larger buffers in order to avoid potential transfer stalls that could reset the entire connection; therefore, MPTCP may have a higher tool in memory resources compared to TCP.

The last subplot is the HTTP traffic flows. The out-of-order delay recorded looks like the short-lived flows (95% of the time below the $100ms$ mark) as expected, considering that HTTP traffic is mostly short-lived flows, see Section 6.3.2. In general we saw that velocity does have a small impact on the delay caused by reordering the packets, but not significantly since the underlying connection used is cellular. In Section 6.4.3.2 we show the real impact that high moving dynamics may caused to an MPTCP connection.

6.4.3.2 RTT and Path loss

Figure 6.13 shows the complementary cumulative distribution of the RTT on the cellular path and Figure 6.14 shows the same for the WiFi path in our MPTCP experiments. Just like in Section 6.4.3.1, we include a pseudo-spectrogram per subfigure that shows where the RTT values predominate. Table 6.3 shows a summary of the RTT and Path Loss values recorded for the short and long-lived flows; Table 6.4 shows the same for the very long, continuous and HTTP traffic flows.

For short-lived flows, the probability that the RTT recorded on both the cellular and the WiFi path stays below the $100ms$ mark is 90% on the 25mph and 45mph experiments. The transmissions at 65mph suffer the most in terms of RTT with slightly worse performance as shown in their respective spectrogram plots in both paths. From Table 6.3 we see that both the cellular and the WiFi path have negligible loss, except for the WiFi path at 65mph which is clear that it has an impact. The RTT values for short-lived flows on Table 6.3 are better for the cellular path than on the WiFi; this explains why the MPTCP scheduler allocated more packets on that path.

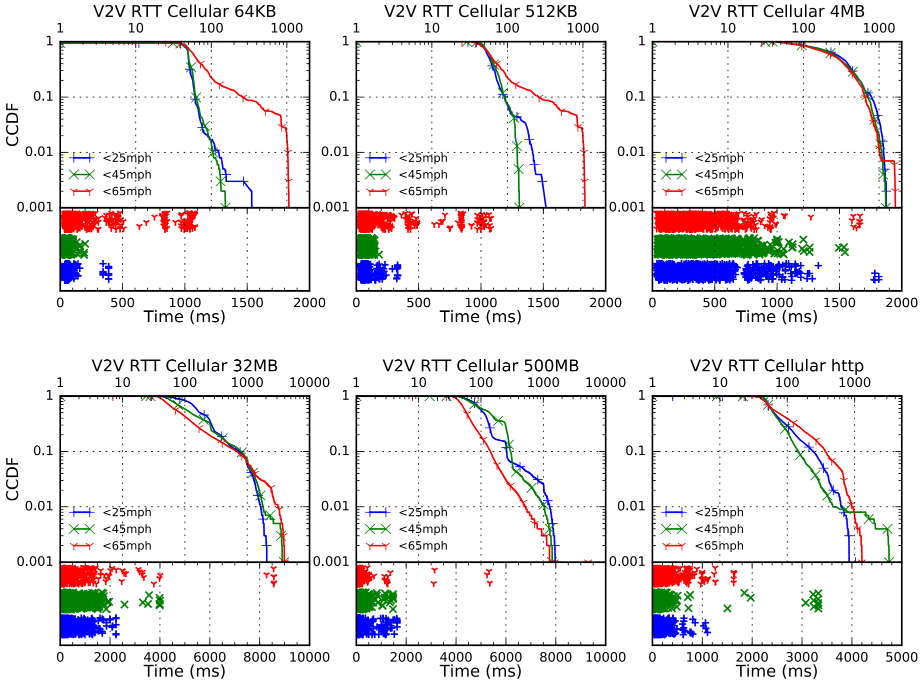


Figure 6.13: CCDF of RTT on the cellular path

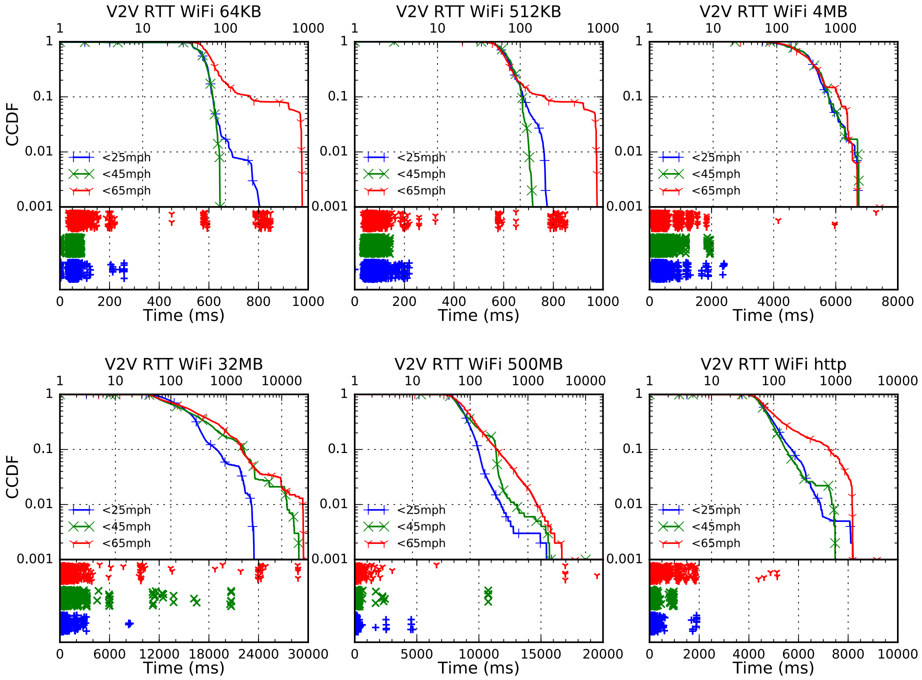


Figure 6.14: CCDF of RTT on the WiFi path

Cellular RTT (ms)

Speed	64KB	512KB	4MB	32MB
25mph	52.33 ± 8.44	75.30 ± 28.48	264.13 ± 146.67	170.04 ± 70.33
45mph	52.73 ± 6.85	70.86 ± 21.42	252.84 ± 139.75	139.06 ± 81.48
65mph	55.82 ± 16.16	82.72 ± 35.66	251.28 ± 138.09	98.91 ± 68.37

WiFi RTT (ms)

Speed	64KB	512KB	4MB	32MB
25mph	81.50 ± 23.56	104.73 ± 41.62	312.10 ± 126.31	207.35 ± 86.88
45mph	77.72 ± 18.38	154.93 ± 64.63	288.04 ± 135.53	250.44 ± 291.09
65mph	110.86 ± 48.09	91.54 ± 39.13	233.87 ± 109.00	335.77 ± 284.49

Cellular Loss (%)

Speed	64KB	512KB	4MB	32MB
25mph	0.04 ± 0.26	0.05 ± 0.14	1.89 ± 2.72	3.16 ± 2.16
45mph	0.03 ± 0.25	0.04 ± 0.19	1.29 ± 2.54	2.72 ± 3.27
65mph	0.23 ± 0.71	0.64 ± 2.52	3.97 ± 3.54	2.33 ± 3.65

WiFi Loss (%)

Speed	64KB	512KB	4MB	32MB
25mph	–	–	1.01 ± 1.50	3.93 ± 7.12
45mph	–	0.10 ± 0.17	2.04 ± 5.17	3.48 ± 3.40
65mph	1.97 ± 2.16	5.05 ± 7.59	1.41 ± 3.75	2.92 ± 3.06

Table 6.3: RTT and Path Loss values on short and long-lived flows

	Cellular RTT (ms)		**Cellular Loss (%)**	
Speed	HTTP	500MB	HTTP	500MB
25mph	56.51 ± 12.26	205.47 ± 63.37	0.10 ± 0.69	3.80 ± 3.24
45mph	56.52 ± 10.05	207.15 ± 57.95	0.05 ± 0.60	0.86 ± 0.74
65mph	60.42 ± 19.12	69.50 ± 43.30	0.33 ± 1.71	0.18 ± 0.00

	WiFi RTT (ms)		**WiFi Loss (%)**	
Speed	HTTP	500MB	HTTP	500MB
25mph	50.27 ± 7.64	237.00 ± 71.47	0.04 ± 0.53	3.75 ± 6.37
45mph	51.04 ± 7.06	268.33 ± 80.83	0.02 ± 0.55	5.10 ± 7.45
65mph	47.02 ± 8.30	146.80 ± 289.80	0.99 ± 4.10	0.25

Table 6.4: RTT and Path Loss values on 500MB and HTTP traffic

For long-lived flows, specifically the 32MB flows, we see the same story as in short flows, except that there are larger losses in Table 6.3; this is expected as these connections expose longer to the vehicle's speed. There is about 80% probability that the RTT falls below the $500ms$ mark for the cellular path, and the same probability for the WiFi path's RTT to be within $1s$. The velocity has the greatest impact on RTT for MPTCP connections because the MTPCP scheduler selects the path to schedule packets based on the RTT value. This also explains why the traffic offloading stays below 30% in all cases, see Section 6.4.2. In the 4MB flows, a special case in our analysis, we can see that there is little distinction on the RTT performance among the velocities but they are all equally impacted. The only differentiating factor is, therefore, the T-Mobile download performance shown in Table 6.2, the coverage hole.

The very long, continuous transmission of 500MB follows a similar pattern to the rest of the experiments gradually degrading with the velocity increase. However, there are better probabilities to have RTT values below the $300ms$ mark: about 90%, 10% less than 32MB flows. This is significantly better, even if the 500MB flows are exposed longer to the vehicle's movement. The reason behind this result is shown in the spectrogram: the RTT values for the 500MB transfers are more compacted with fewer outliers compared to the 32MB transfers. Moreover, the 500MB flow has better loss values than the 32MB flows in Table 6.4. Overall, it is clear that the 500MB flows have smoother transmissions compared to the 32MB flows; however, the difference is not significant and the pattern is consistent with the rest of the experiments. The HTTP traffic has also similar behavior in RTT CCDF compared to short-flows, however, velocity does have a larger impact. We recorded similar RTT values and losses as well in Table 6.4 compared to those in short-lived flows in Table 6.3.

In general, we observed that the vehicle's velocity does have a significant impact on the RTT recorded on either path of the MPTCP connection; there is also an impact on the recorded losses as well, but not so significant. Seeing average RTT values of at most $400ms$ with probabilities of around 80% below the $1s$ mark in the worst case, we find it difficult to justify the use of MPTCP connections on applications that are delay sensitive. However, for applications that can relax the latency constraints, the loss in latency performance is justified by the gains in traffic offload.

6.5 Analysis

In this section we tie together the results in Section 6.4 and deliver our recommendations regarding the use of MTPCP in VANET scenarios. In Section 6.4.1 we found that the bandwidth degrades progressively with the increase in the vehicle's velocity; more pronounced on the larger velocities than on the lower ones. In Section 6.4.3 we discovered that velocity has stronger degradation in terms of latency. In Section 6.4.2 we identified that seamless handovers and offloading are the stronger advantages of MPTCP in the wireless context. These results are expected because of the large dynamics in wireless factors. The main goal of this study is to understand if MPTCP is viable for V2V vehicular scenarios, and if it is, how well does it perform compared to the state-of-the-art.

In general we found that MPTCP does work under V2V VANET scenarios. We found that MPTCP does not greedily take the residual bandwidth available to its TCP subflows that the network provides, as one could initially expect; in fact, it tries to be fair, as a normal TCP flow would be. In our experiments the bandwidth performance of MPTCP matches or stays around those in the SP-TCP transmissions, that is the ideal case for MPTCP according to its designers. We also found that long-lived flows are exposed longer to the dynamic moving factors of the vehicle; therefore, we see larger losses and latency issues there. As a result, we found that MPTCP is currently not ideal for delay sensitive application, such as voice-over-IP or live video conferencing.

Multipath networks are fertile field for MTPCP connections, especially in moving scenarios where TCP connections cannot maintain end-to-end connectivity as MPTCP. For these scenarios, we found that the major advantage of MPTCP is traffic offloading. In scenarios with large vehicle movement, we found that MPTCP is still able to offload a considerable large portion of its load, between 20% to 30% approximately. This feature could help decrease the costs of a cellular provider; also, in rush hour scenarios, it is possible to shift the path away from congested or failing cellular networks to the path provided by a peer vehicle seamlessly. We saw that offloading is viable and practical even at velocities as large as 65mph, at the cost of latency degradation. In summary, we found that MPTCP does work well under VANET scenarios, it is able to maintain connectivity, match SP-TCP bandwidth performance and offload traffic onto other paths seamlessly when available; the trade-off that we identified is on latency.

Figures 6.6 to 6.9 include some issues that may adversely affect the performance of MPTCP under large movement. The first is identified in Section 6.4.1.2 regarding the 4MB flows; it is possible that negligible offloading is recorded if the cellular network is failing or there exists coverage holes in the neighborhood. We also found that if there is a link failure in the middle of the network that is included in the main path, the whole MPTCP connection stalls if other paths have not been setup yet. An example of this scenario could be when an MPTCP socket opens a new connection and the SYN packet is lost or never delivered. The first issue would not apply to TCP since it does not use multiple flows in its end-to-end connection; MPTCP downgrades to TCP in this case. For the second issue, both TCP and MPTCP would naturally fail, as both are end-to-end connections; the problem is that MPTCP does not take advantage of the other interfaces available at its disposal to start the flow. In this study we identified this issue, but we left as a future study item.

6.6 Related Work

This paper studies the performance of MPTCP in a VANET on V2V configuration. We based the presentation of our work on Chen's, et. al., in [20]; we adopted their study methodology, and we feel our work is complementary to theirs. Bychkovsky, et. al., studies the use of WiFi on VANETs in real-life scenarios in [16]; this work is very close to ours but we use MPTCP as the main Transport protocol. Arzani, et. al., studies path characteristics and scheduling policies to evaluate MPTCP performance in [7]; Paasch, et. al., gives an insight on MPTCP schedules in [102]; Jiang, et. al., studies the concept of bufferbloat in cellular connections in [68], Ferlin-Oliveira, et. al., and Chen, et. al., do it over MPTCP [42, 21]; Li, et. al., takes a look into the role of receive buffers in [84]; Baidya, et. al., goes one step further into proposing an algorithm to improve the issue of small receive buffer degradation of MPTCP in [8]; these last works explore fundamental concepts that impacted our MPTCP performance analysis under VANET scenarios. Lee, et. al., also study MPTCP under mobility in [81]; they propose to integrate a mobility context state, i.e., walking, driving, onto the path selection; our study uses the MPTCP protocol as proposed by the standard [43]. Deng, et. al., measures the MPTCP performance over multiple interfaces in [30], but there is no mobility involved. Ferlin, et. al., takes a look at the role of path selection and scheduling in [41] over heterogeneous wireless networks; Nguyen, et. al., evaluate MPTCP in terms of throughput and load sharing in [98]; Zhou, et. al., studies cooperative MPTCP over LTE interfaces in [131], just like our V2V test bed; in all these works, there is no mobility involved. Riaciu, et. al., studies the performance of MPTCP under mobility in [107]; however, their work only focuses on indoor/outdoor simulations while we focus on real-life, outdoor VANET scenarios. Paasch, et. al., studies the MPTCP performance over mobile/WiFi handovers in [101]; but their work focuses on controlled mobility and energy concepts. Chirwa, et. al., applies MPTCP to a platoon of cooperating drones in [22]; this work studies a scenario very close to the VANET but it is under simulation. Han, et. al., studies the impact of MPTCP over mobile web applications in [52]; Pollalis, el. al., and Hampel, et. al., propose in [104, 51] a proxy server to allow non-MPTCP speakers engage into multipath connections. Our test bed uses HTTP traffic but under movement without proxies in our architecture.

6.7 Relevance to the Thesis

The work on this chapter presents the experiments we performed on a VANET configured as a V2V, where vehicles shared their cellular connections to connect to the Internet. Such collaboration is meant to maintain service continuity on applications that leverage MPTCP to maintain multiple connections. We observed a single vehicle within a platoon of two in order to study how well MPTCP behaves when it offloads some of its traffic onto the peer vehicle; our observations apply on the other vehicle as well. In general, we found that vehicles can offload traffic between 20% to 30%, approximately, among themselves at speeds around of 65mph, 105km/h. These results demonstrate that service continuity with multiple paths provided by a platoon of vehicles is possible and viable, which demonstrates this thesis goals in a highly dynamic scenario.

CHAPTER 7

MPTCP Path Selection using CapProbe

7.1 Introduction

It is not uncommon that current mobile devices can reach the Internet through WiFi, cellular LTE, Ethernet using proper adapters, or even Bluetooth connections; however, priority is usually given to energy conservation rather than network performance. Many devices, including smartphones, allow at most one active network interface to maintain Internet connectivity. Continuous, long-lived end-to-end communications such as video transfers or VPN sessions cannot rely on TCP without braking the connection when the device is on the move. Multipath TCP is a TCP backward-compatible transport protocol helps in this scenarios by leveraging the multiple interfaces in a mobile device to perform handovers and maintain these continuous sessions running. When the mobile device enters the coverage of a new network, MPTCP may add a new path to support or substitute a currently running one through a handover. Unfortunately, network performance may degrade if the mobile device hands over onto a path with poor characteristics, i.e., one that depicts high latency or low capacity. Currently the MPTCP protocol does not perform proper path selection on the networks it finds. Although WiFi networks are preferred over cellular ones due to energy efficiency, the mobile device should not handover at the expense of significant performance decrease or disconnection.

This work addresses the MPTCP path selection based on probing the newly discovered paths to qualify them and revert a handover if the path is deemed as low unsatisfactory. To measure the paths, we use CapProbe, a path capacity measurement, and identify metrics to help asses the path characteristics. We propose algorithms that measure and select paths so that the mobile device may hand over out of a bad paths and resume continuous network connectivity without user interven-

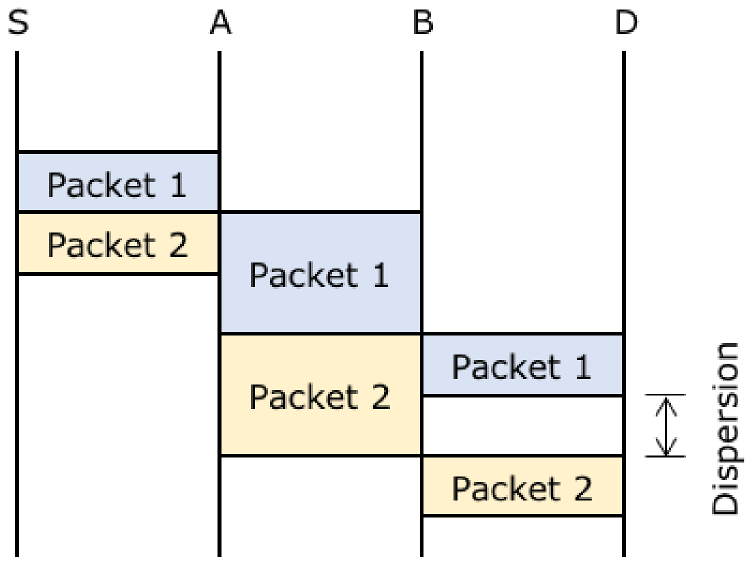


Figure 7.1: Time dispersion occurs after link A-B opens up a gap between the back-to-back packets due the link's low capacity.

tion. We present an analysis of the metrics selected and run experiments on scenarios where the metrics are relevant, i.e., paths with poor characteristics.

7.2 Background

7.2.1 CapProbe

The capacity of an Internet path is the largest bandwidth a path may offer to an end-to-end connection. This capacity is upper-bounded by the narrowest link in that path and knowledge of it is desirable in many scenarios; for example, ISPs may have a better idea about the path characteristics of their links, routing algorithms may redirect flows away from congested zones, overlay networks may reconfigure their connectivity trees to optimize throughput, etc.

CapProbe [71] is a technique that measures the capacity in a path using two techniques. First, by using a series of probes generated by the Packet Pair algorithm [76, 15]; the probes' latency is measured and the capacity is statistically estimated. Second, by using the probe's time *dispersion*; the time dispersion of a packet is the time elongation suffered during its transmission through

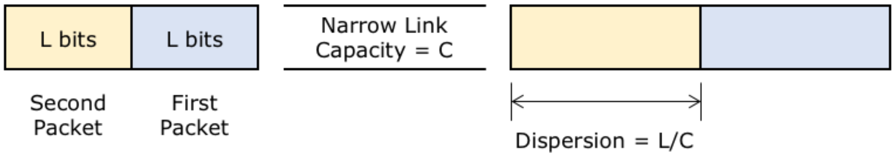


Figure 7.2: Relationship between size, dispersion and capacity

a narrow link in the middle of its path. Figure 7.1 shows the dispersion concept in a network time diagram; note the increase in time required to deliver a packet through a narrow link A-B. When a packet pair is delivered through this path back-to-back and assuming that these packets are always queued together, the time gap between the dispersed packets at the destination can be used to estimate the path's narrowest capacity, regardless of the capacities of the other links along the path. Figure 7.2 shows the relationship between the measured dispersion and the capacity for a packet of size L; when a packet pair arrive back-to-back, if there is no queuing in these two packets, then the second packet will depart L/C time units after the first, which is equal to the dispersion demonstrated in Figure 7.1 between the packets [71]. In practice, the queuing assumption may not be valid due to cross-traffic causing inaccurate dispersion measurements that lead to capacity over and under-estimation.

CapProbe addresses this issue by proactively sampling for a packet pair probe that has not suffer any cross-traffic induced delay; a probe that meets this condition has a valid dispersion value that is used to estimate the path's capacity. In a round scheme, it sends a fixed number of packet pair probes and it measures each of the packet's delay in the probe; the sum of these delays is called the *delay sum* of the probe. CapProbe maintains a historical record called the *minimum delay sum* from all the delay sums seen in the run; it also maintains a *minimum delay* for all the first and second sent packets in the probes throughout a round. By comparing the minimum delay sum with the sum of minimum delays, a probe with the correct dispersion value is discovered in a round; this comparison is called the *minimum delay sum condition*. A probe that meets such condition is said to be one that has not suffered any queuing delay due to cross-traffic; the dispersion value in this probe may be used to estimate the path's capacity. Two runs are executed and the capacity is estimated per round; the final capacity estimation reported is the average of the estimated capacities in the two rounds. For further details on the theoretical justification of CapProbe, we direct the reader to [71].

7.3 Problem Illustration

In the context of mobile devices, energy conservation may be prioritized over network performance. Naturally, either MPTCP backup mode or path deactivation may be used; for example, turn off the cellular interface while there is WiFi coverage; handover to cellular when the WiFi coverage disappears. There are situations, however, where such a handover is not desired because it could lead to network disconnection. To illustrate this concept, take the scenario in Figure 7.3 A WiFi access point may be located behind multiple obstacles but it can still be reached by a mobile user within its coverage. In this situation, we say that the MPTCP connection is *trapped* by the WiFi path because even thought the signal is bad, the WiFi network can still be reached, bad performance entails and no handover back to the better path occurs. This situation may lead to a poor connection due to a very lossy channel and if MPTCP operates under backup mode or maintains this path active, bad performance is recorded. Figure 7.4 shows a distinct scenario where the user may be trapped again by a WiFi path but this time due to a low capacity link along the Internet path. Examples of this scenario may be an overloaded WiFi access point at a coffee shop or a WiFi access point connected to an overloaded subnet or a router, or during an Internet outage. In general, even though energy conservation might be priority, often this configuration does not mean that network connectivity must be sacrificed. Therefore, we call the problem of identifying the best path for an MPTCP connection using at most one interface due to energy constrains as *path selection*.

It is clear that a path must be carefully selected before it is made available to the MPTCP scheduler in order to avoid performance degradation or stalling. Note that if MPTCP operates under fullmesh mode on the default scheduler and no path operates under backup mode or all its paths are activated, then the scheduler would simply schedule its packets onto the next available path. However, in mobile devices that prioritize energy conservation this is not the case [6].

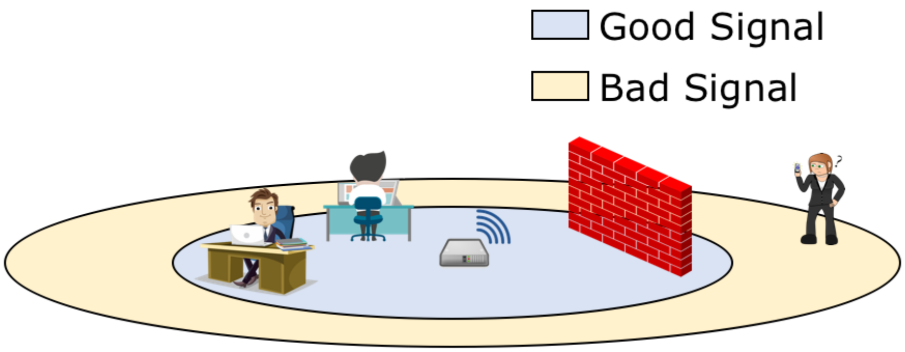


Figure 7.3: A mobile device trapped by a WiFi AP with poor signal.

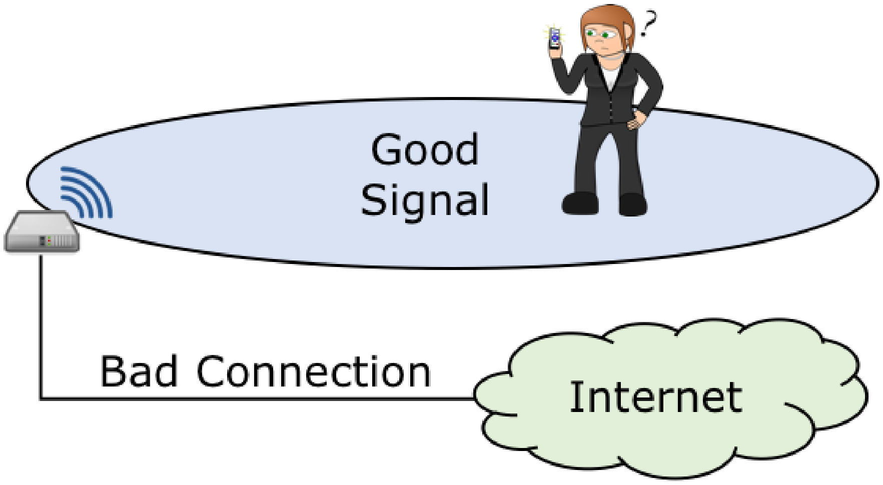


Figure 7.4: A trapped mobile device by a good performing AP with a bad backhaul.

7.4 Metrics

We address the problem in Section 7.3 by measuring the path before it becomes available to MPTCP for handover. This measurement must be fast and it should not overwhelm the path. In Section 7.2.1 we discussed the path capacity concept using CapProbe; our first approach was to measure the capacity of each MPTCP path and select one with the highest capacity while turning off the remaining ones. We found, however, that CapProbe takes a few seconds to converge to a valid capacity value and even though the authors claim that it allows "on-line" mode operation, it is not fast enough to make a path decision. This convergence time is due to the time it takes to find a probe that satisfies the minimum delay sum condition as discussed in Section 7.2.1. Probes that satisfy this condition contain a valid time dispersion that can be used to estimate the path capacity accurately. After valid probes are found, the smallest time dispersion value in those probes is used to estimate the capacity as $C = L/T$, for the corresponding L fixed-size probe and a minimum time dispersion T. Packet pair probing measurements may lead to inaccurate results due to two reasons; first, system clocks with low resolution; and second, interference due from cross traffic. While it is expected that advances in technology lead to better system clocks, the CapProbe authors address the cross-traffic induced delay with their proposed minimum delay sum condition. In the next section we show how all the dispersion measurements, and not only those obtained from probes that satisfy the minimum sum delay conditions may help us estimate the path capacity by sacrificing some accuracy that we do not necessarily need.

7.4.1 Sub-optimal Fast Capacity Estimation C'

In selecting the best paths, we do not need an accurate capacity measurement to take a decision, we simply need an indication of the traffic load to differentiate these paths. We can relax the CapProbe's minimum delay sum condition and take any time dispersion measurement even if it leads to capacity over- and under-estimation. As a result, these dispersion measurments will come from probes that experience queuing delay between and before the packets that is responsible for sub-optimal capacity estimation; such inaccuracies are discussed in Section 2 in [71]. For path selection, this is exactly the behavior that our metric needs to identify. Therefore, we refer to

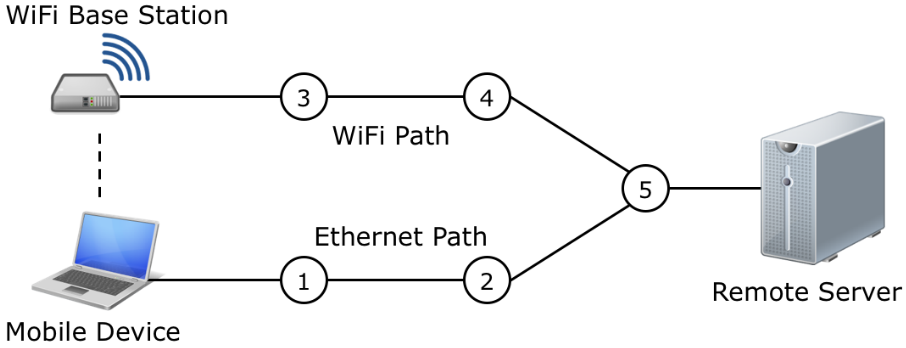


Figure 7.5: Small test bed used to control the capacities of each path; links 1-2 and 3-4 are narrow links.

the sub-optimal capacity estimated by a time dispersion value in a probe that does not satisfy the minimum delay sum condition as *sub-optimal estimated capacity*, or C'. In order to avoid oscillations, we smoothed the time dispersion values just as the Linux Kernel smooths the RTT with an alpha value of $1/8$, before we computed C'.

Figure 7.6 shows the behavior of C' on an Ethernet path as a function of the narrowest link transmission rate. To test our metrics, we used a small test bed shown in Figure 7.5 that includes two paths, WiFi and Ethernet, between a mobile device and a remote server that engage into an MPTCP connection[1]. In order to control the capacity on the Ethernet path, we use the link 1-2 to control its transmission rate[2], this is path's narrowest link; the same goes for link 3-4 for the WiFi path. To control the WiFi channel quality, we use the WiFi base station transmission power to enhance or weaken the signal. In the figure, the blue curve represents the estimated capacity while the red curve represents the dispersion ratio metric; we introduce the latter metric in Section 7.4.2. We first notice from the figure that the estimated capacity C' follows a downward trend as we tighten the narrow link, becoming more accurate as the transmission rate becomes narrower. At high transmission rates, the estimate is significantly off, particularly at transmission rates higher than 50Mbps. We attribute this reasonable accuracy at narrow links to queuing induced delays on the measurement that overshadow the clock inaccuracies while at high data rates, these inaccuracies have a larger impact. Although C' is not as accurate as CapProbe's capacity estimation, this

[1] To generate traffic, we use the *curl* Linux utility to download a 1.5GB Ubuntu image file.

[2] We use the Linux *tc* queue discipline HTB and NETEM to control the transmission rate

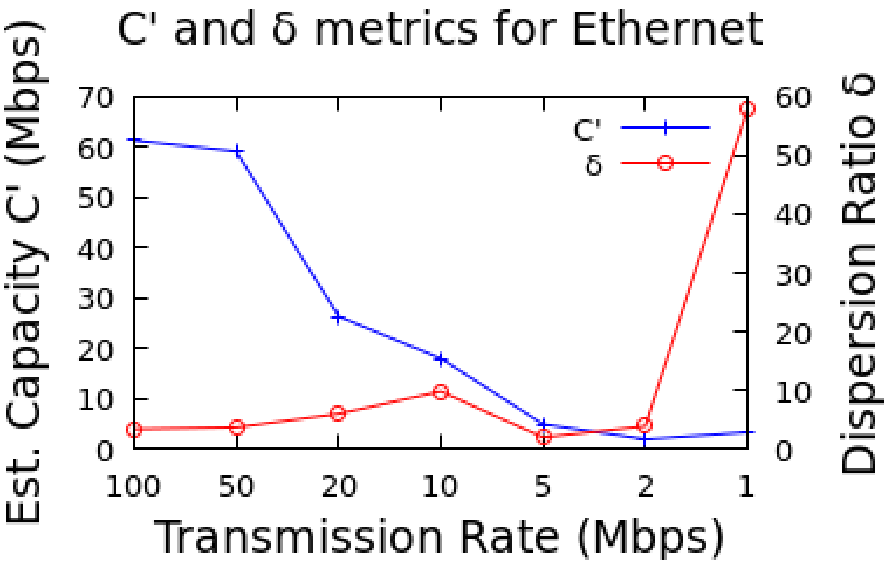


Figure 7.6: Ethernet Path

metric gives us a very good indication of the path's quality.

Figure 7.7 shows the behavior of C' on a WiFi path but this time as a function of the base station signal at the mobile station. We see a similar downward trend as the signal strength decreases; this is due to the dynamic data rate in IEEE 802.11. At weaker signal levels, the transmission rate decreases due to less granularity in the signal modulation used to ensured the reception of a transmission. The numbers at each point represent the measured maximum throughput at each signal value[3], giving us an indication that C' becomes more accurate as the signal fades, just like in Figure 7.6.

7.4.2 Dispersion Ratio δ

The C' metric works well to identify bad quality paths. However, it some instances it could be significantly inaccurate, especially at high transmission data rates. We need another supporting metric that can identify a path that has degrade in quality due congestion effects. CapProbe already measures the time dispersion; it also maintains the minimum time dispersion value it has seen in a round. This minimum value represents the best dispersion measured that satisfies the minimum

[3] We use the Linux tool *ifstat* to capture the overall throughput on the interface.

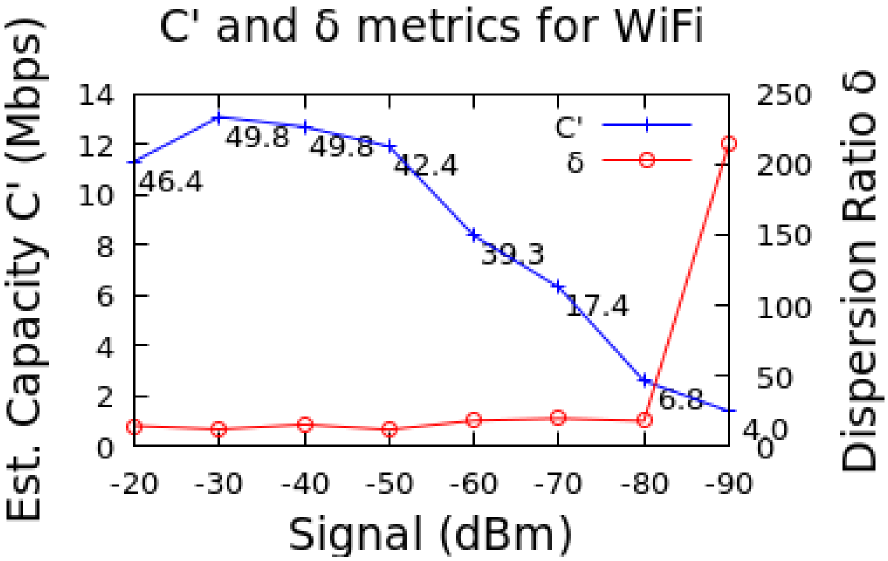


Figure 7.7: WiFi Path

delay sum condition so far; thus, it is the best this path can do. If we take the ratio of the smoothed dispersion value to the minimum dispersion value from CapProbe that we have obtained so far, we measure how far worse the quality of a path is from its current ideal; we call this *dispersion ratio* δ.

Figures 7.6 and 7.7 also show the dispersion ratio on both the Ethernet and WiFi path under the red curve. This metric reacts particularly well to situations with large congestion; that is, the probing is able to identify some small dispersion values that approximate to the real capacity but most probes experience severe queuing delay leading to high dispersion values. When this happens, the ratio increases. On the other hand, when there is little congestion, the smoothed time dispersion would be close to the minimum dispersion value seen so far most of the time, thus decreasing the ratio δ. Together, the C' and δ metrics are able to identify a path with low capacity that is significantly worse than its ideal performance. We call a path that depict such characteristics a *bad path*.

7.5 Algorithm

We address the problem of path selection with two algorithms introduced in this section.

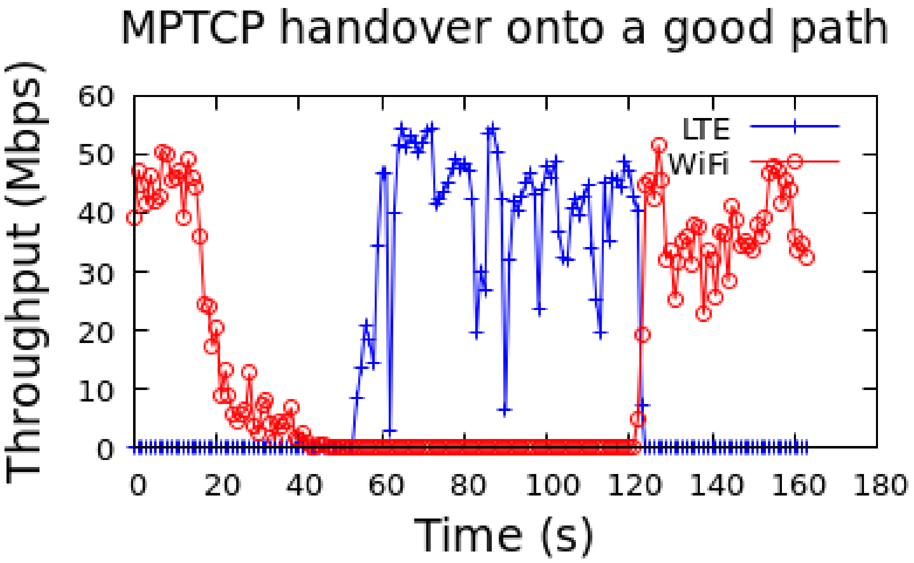


Figure 7.8: A good MPTCP handover

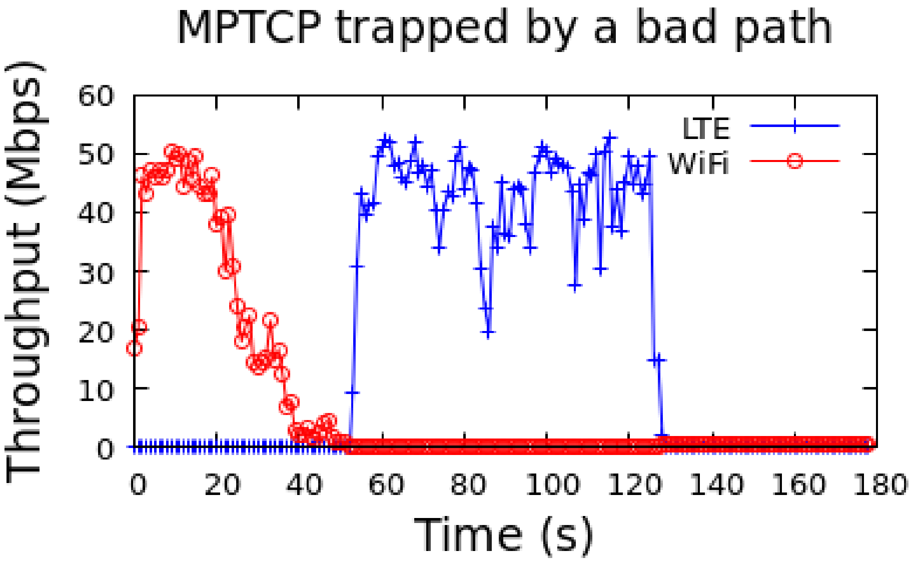


Figure 7.9: A bad MPTCP handover

C'	δ	RTT	Path Quality	Description
			✓	Path is good
		✗	✓	Low responsive path
	✗		✓	Isolated congestion indication
	✗	✗	✗	Strong congestion indication
✗			✓	Sudden loss in capacity
✗		✗	✗	Stable low capacity
✗	✗		✗	Consistent bad path quality
✗	✗	✗	✗	Bad path

Table 7.1: Fundamental description of each combination of set metrics.

Algorithm 1: Path Grade

```
for each path $p$ do
    if $C' < C\_THR$ and $\delta > D\_THR$ then
        mark $p$ as bad path
    else if $C' < C\_THR$ and $srtt > SRTT\_THR$ then
        mark $p$ as bad path
    else if $\delta > D\_THR$ and $srtt > SRTT\_THR$ then
        mark $p$ as bad path
    else
        mark $p$ as good path
    end if
end for
```

Table 7.1 shows the path state meaning using the combination of C' and δ metrics combined with the smoothed RTT. Obviously, the best case is when all metrics are fine. If only RTT lags, the path has high latency, for example, links depicting high bandwidth-delay product such as satellite; however, we set a limit on how unresponsive a path can be; this limit is the threshold $SRTT\_THR$. When C' is low or δ is high we have some indication of path quality degradation but it may be an isolated case. However, when both metric indicate quality decrease, it is a strong indication that the path is not improving and it should be marked as a bad path. A low C' value is one that crosses below the C_THR and a high δ values is one that passes above the D_THR. High δ and SRTT values indicate congestion but low C' and high SRTT may not, it could just simply be a low capacity path. In both cases, however, the path is undesirable and it should not be prioritized for selection. Lastly, when all metrics report bad quality, the path is marked as bad.

Algorithm 2 initiates CapProbe measurements. Every γ time period, it updates the path's statistics on all the available paths; then each path is graded based on their statistics and the path handover is performed if the path quality had decreased. Algorithm 1 implements the concepts in Table 7.1 in order to properly classify the path's quality.

Algorithm 2: Path Selection

```
start CapProbe
while every time period $\gamma$ do
    for each path $p$ do
        $C' \leftarrow L/disp$
        $\delta \leftarrow sdisp/mindisp$
        $srtt\_ratio \leftarrow wnd\_avg(srtt)$
    end for
    Path Grade
    if current active path is marked bad $N$ times then
        Handover to the backup path
    end if
    sleep for $\gamma$ time
end while
```

The metrics declare a path if they meet a predefined threshold. For C', we have threshold C_THR and it is set to 1Mbps; that is when the estimated capacity falls below this threshold, we have an indication of a bad path. For δ, we have D_THR; we set it to a ratio value of 100 in order to account the bursty behavior in WiFi. This value was chosen solely on our experiment

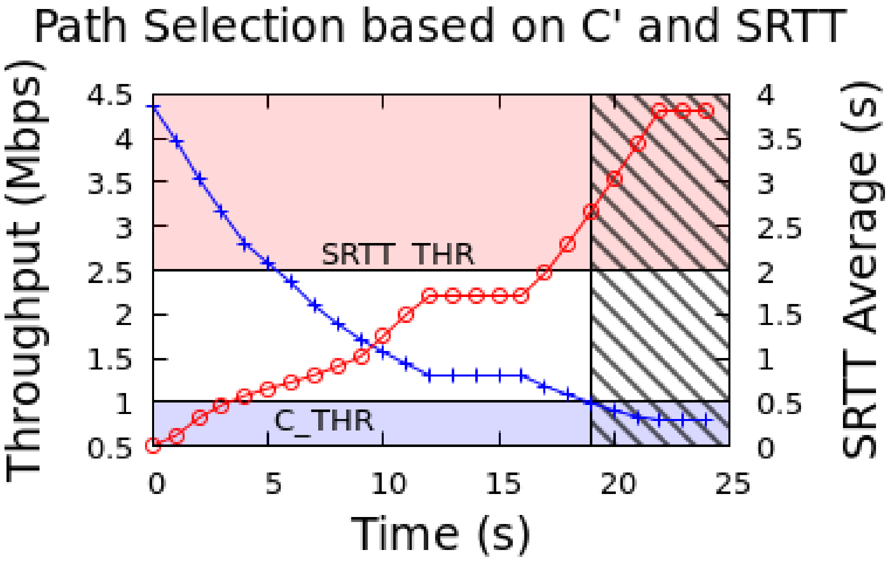


Figure 7.10: Metrics reacting to a bad path

observations. The SRTT has the threshold $SRTT\_THR$, and we set it to 2000ms. Based on our observations, smoothed RTT values of 2s depict very high latency paths. We use this threshold to corroborate another one, e.g., C_THR or D_THR, that specializes in congestion to make the following statement valid: "if there is congestion and the path is unresponsive, mark this path as bad". Using a N second rule, five in our experiments, we avoid oscillations and assured the path is indeed a bad one.

7.6 Evaluation

For our experiments, we use two Lenovo W520 laptop running MPTCP v0.92 on Ubuntu 16.04 LTE Linux as the mobile device and the remote server when they engage into MPTCP connections. The mobile host connects to the Internet via a WiFi interface; it also uses LTE via an USB-tethered iPhone SE through AT&T. We configured the mobile device to use the WiFi path whenever possible and setting the LTE path as backup, just as smartphones are configured. The mobile device connects to an remote web server to download web objects in order to generate traffic. Our algorithms operate at all times controlling the WiFi interface only; we assume that the LTE path will always be there. There is no point in controlling the LTE since the WiFi path is always preferred

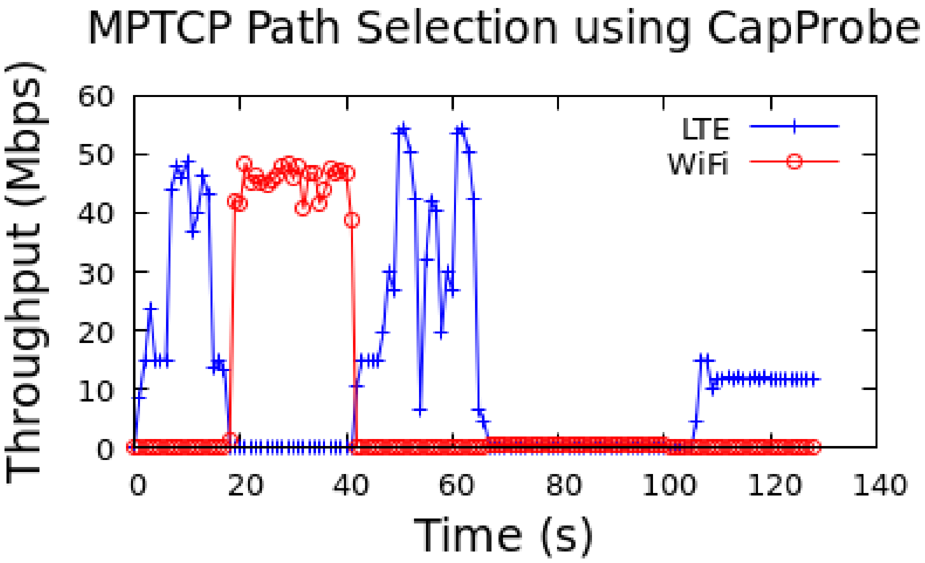


Figure 7.11: Recovery of a bad handover

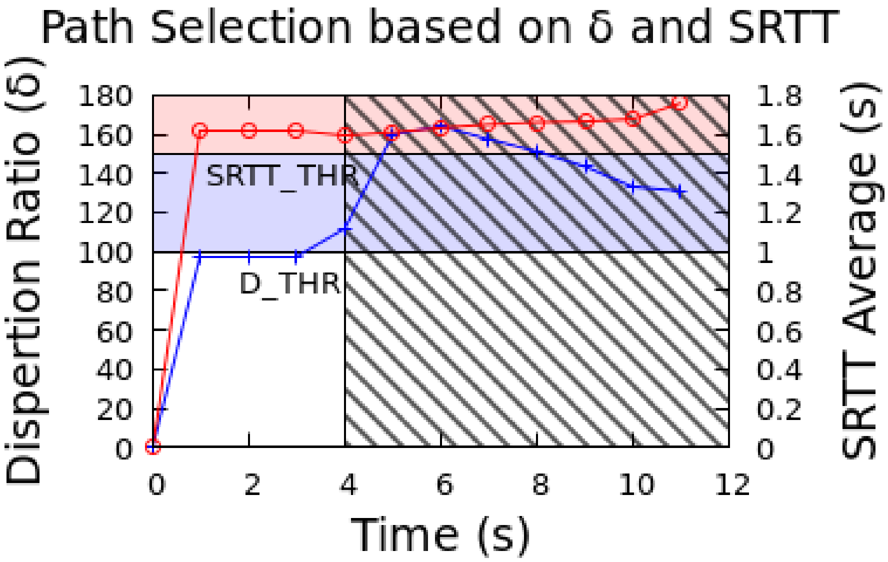


Figure 7.12: Metrics detecting a bad path

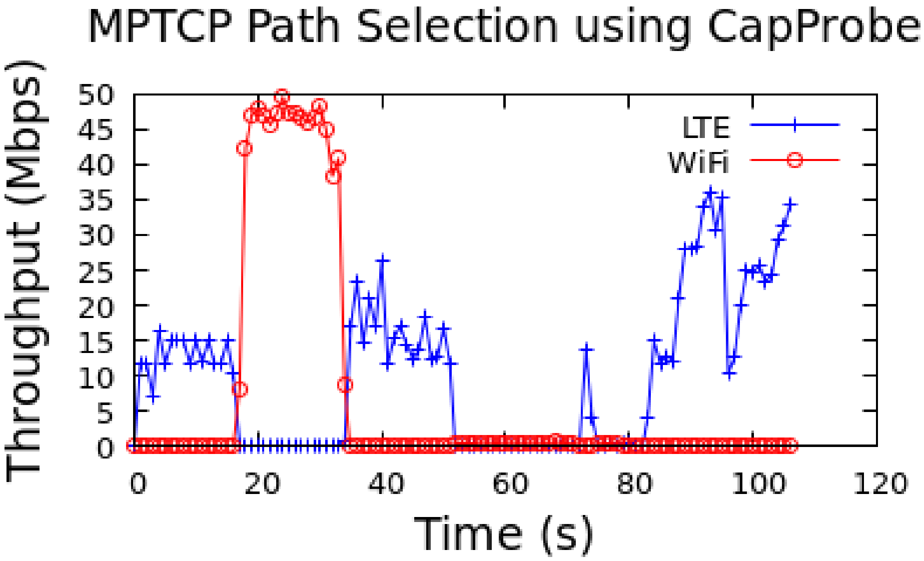


Figure 7.13: Recovery of a bad handover

unless our algorithms find it to be a bad path; when there is only LTE, our algorithms are not relevant to this case. We run multiple experiments at a local coffee shop as well as in a residential home wireless network; we found little differences in both scenarios considering that in the former there is high contention for the WiFi access point while in the latter there is high interference from nearby WiFi access points. We describe two representative instances of our algorithm at work due to lack of space.

Figure 7.11 shows the throughput progression of our first MPTCP connection instance when it hands over between LTE and WiFi. The connection first starts on LTE (shown in blue), then we turn on the WiFi interface (shown in red) and MPTCP hands over. After about twenty seconds we walk away until the WiFi interface loses connection to the base station and it hands over back onto LTE. During the next seconds we carefully walked back into the coverage of WiFi but staying far away enough to be on the edge in order to illustrate the case shown in Figure 7.3. The mobile device reconnects to the WiFi base station just before the 70s mark; however, little throughput is achieved and the connection hands over back to LTE after the 100s mark. Figure 7.10 shows the metrics that triggered the handover to LTE due to WiFi path failure in Figure 7.11. The blue curve shows the progression of the estimated capacity C' while the red curve shows the SRTT on the WiFi path; this correspond to line 4 in Algorithm 1. The colored shaded areas represents the where each metric meet their respective thresholds while the crossed area represents the case when the two metrics declare the path as bad. Note how the capacity estimation is converging to around 1-1.5Mbps after the 12s mark and later crosses its threshold at the 19s mark. The SRTT keeps increasing as the connection struggles to push packets out of the interface competing with the other users sharing the base station's bandwidth. The SRTT meets its threshold one second before C', at the 18s mark. After about nineteen seconds of measurements, both metrics declare the path as bad and after five consecutive declarations, the handover produces. This happens around the 100s mark in Figure 7.11.

Figure 7.13 shows another instance like our previous example. Figure 7.12 shows the metrics that triggered the handover to LTE when the WiFi path fails, but this time the metrics that triggered the handover are dispersion delay δ and SRTT. MPTCP hands over after the 50s mark but it encounters a bad path almost immediately. Figure 7.12 shows that after about four seconds of measurements, δ and SRTT declare the path bad; our algorithm waits for another five seconds of continuous bad path declarations before it shuts it down; in the figure we can see that this happened at the 9s mark. In Figure 7.13 this handover should happen in the middle of the 60s; but the connection struggles to recover from all the losses occurred by the previous handover into the WiFi path. We see a spike in throughput around the 70s mark, as retransmissions occur, the LTE path picks up throughput just after the 80s mark.

In the two examples shown, our algorithm is able to detect a bad path almost immediately; this is thanks to the CapProbe-based metrics used by our algorithm. Our algorithm waits for five bad-path declarations in order to ensure the path selection is valid; five declarations mean five seconds since we collect measurements every second. We also noticed that the backup pack may struggle to recover throughput after our algorithm issues a handover; we attribute this issue to the packets lost and retransmissions required since the mobile device uses only one path. An MPTCP connection under the current MPTCP implementation would stuck or stall stall the throughput and no handover would be triggered; in the former case, because throughput is still achieved and the path still is active, and on the latter because even though the base station signal is strong, it fools the MPTCP path manager into believing that the path is good, but a bad link behind the base station is actually the culprit for the path's under-performance. MPTCP is an end-to-end protocol so

	Ethernet	WiFi	LTE
Overhead in Kbps	150.5	143.5	77.3

Table 7.2: CapProbe overhead imposed

scenarios like this are beyond its scope; nevertheless, it detriments its performance; our algorithm helps in mitigating that.

Finally we show the overhead that CapProbe imposes on the path. Even though the only relevant measurement for our algorithm is on the WiFi, we include the CapProbe's toll on the other interfaces as a comparison reference. To run CapProbe, and our metrics, takes 143.5Kbps on average in our measurements. Compared with normal WiFi transmission rates in the high tens of megabits per second, this overhead is negligible.

7.7 Conclusion

This paper proposes two algorithms aimed at enhancing the MPTCP Path Manager at better selecting paths for the MPTCP scheduler use. Mobile devices often focus on saving energy rather that seeking higher network performance in order to extend the device's battery life; therefore, some paths are configured as backup or shut down leaving the device to only one interface to operate on. The current MPTCP Path Manager informs the MPTCP core about the availability of a new path, but if the path has bad quality, the connection could degenerate or even collapse and stall, leading to bad performance; we refer to this event as *trapping* the connection. CapProbe is a metric that measures the path's capacity and can be used to qualify the paths, but it takes time to converge to meaningful results. Leveraging on CapProbe partial measurements, this work identifies two key metrics that can be used to immediately take a path selection decision. Our proposed algorithm uses the sub-optimal estimated capacity C' and the time dispersion ratio δ to qualify a path before it is offered to the MPTCP core. Our results demonstrates that our algorithm can qualify paths immediately and within seconds issue a handover onto a backup path. The main contribution is an automated recovery of an MPTCP path due to a bad handover and the return to former network performance levels without user intervention.

7.8 Relevance to the Thesis

CapProbe and the metrics identified in this work combine to address the problem of path selection and handovers when multiple wireless networks are present. This thesis aims at providing service continuity but when the mobile device leave the coverage area in its wireless interference, a handover is produced. When the mobile device enters the coverage of other candidate wireless networks, it is important to determine if the handover should occur; it may be possible that the handover is not beneficial if the wireless network is lossy and provide bad paths. The algorithms provided in this work provide algorithms that help on the path selection problem; thus, maintaining service continuity in the system.

CHAPTER 8

Vehicular Network Operating on a Cognitive Network

8.1 Introduction

The Industry, Scientific, and Medical (ISM) band, is a portion of the electromagnetic spectrum that is freely available for anyone and any wireless device for use, without a concession fee and for any industrial, scientific, or medical purpose [60]. Thanks to this band, technologies have flourished in very diverse sectors such as communication, mobile health, logistic services, personal area communications with Bluetooth™ and cordless phones, among others [91]. The regulatory body that manages the wireless spectrum use in the United States is the Federal Communications Commission, or FCC [40]; it "regulates interstate and international communication by radio, television, wire, satellite and cable in all the United States." One of the most important FCC regulations is regarding how much power can a device use to send a signal in order to communicate with another wireless device, [39]; this is an important factor that contributes to wireless interference.

In the context of vehicular networks, VANETs, WiFi access points may be deployed as Roadside Units, or WiFi RSUs, which are mobile wireless access points strategically located to assist a VANET platoon. VANETs may also use a freely available WiFi metropolitan infrastructure, like the Time Warner Cable WiFi network in the city of Los Angeles described in Chapters 5 and 6. If we have a better understanding about the current WiFi ISM band's quality then mobile devices in VANETs can reconfigure their wireless interfaces to establish communications at portions of the spectrum that are the least busy, i.e., that have less interference from surrounding devices. This will allow the network to perform better with higher response time.

VANETs are also an ideal candidate for environmental and pollution monitoring due to their high mobility and complex routes in urban scenarios. Rather than monitoring at fixed locations, vehicles may collect pollution samples at distinct areas within a region and deliver them to a remote center for further analysis. Data collected by mobile monitors may help alerting the appropriate environmental agencies of possible dangerous situations hazardous to both humans and the environment.

The goals of this project are, first, understand the impact of WiFi interference of neighboring residential and commercial WiFi access points to WiFi RSUs assisting vehicles operating on a VANET; second, identify which ISM spectrum band is best for a WiFi RSU to use or which metropolitan WiFi access point has the least interference so it can benefit the service for passing vehicles; and third, demonstrate the concept of environmental and pollution monitoring as an application for VANET systems. To these goals we built a test bed introduced in Subsection 2.6 and detailed in Section 8.3.1, capable of measuring the interference that exists on the ISM band. Also we built a separate pollution monitoring test bed introduced in Section 8.3.2, independent of the communication test bed before, that collects and delivers environmental samples to a remote system presented in Section 8.3.3. We embark on the task of measuring wireless interference and pollution data by either walking and driving in order to understand the wireless spectrum characteristics; for the environmental data, our test bed simply senses and collects data that is delivered to a remote server for further processing.

8.2 Pollutant Gases

In order to select which gas sensors are appropriate for our test bed, we investigate what pollutants are present in the areas that we plan to measure. As a join collaboration between the University of California, Los Angeles, and the Universidad Autónoma de Baja California, campus Mexicali, we took upon ourselves to measure the air quality on the cities of Los Angeles, USA, and Mexicali, Baja California, México.

The city of Los Angeles is a major metropolis located in the southern half of the state of California. According to the US Census Bureau, it has an approximate population size of 3.9 million[118] people and it sits in a county with estimate of 7.8 million[18] vehicles circulating on its roads. Mexicali is the state capital of Baja California, the northern-most state of México that shares a border with California. According to data from INEGI, [59], the total population in Mexicali is 936,826 people; it is an industrial and agricultural complex for the state of Baja California, [111].

For this project, as a collaboration effort, we decided to *include in our CogNet/VANET a sensor package that collects pollution information around the areas where we perform our spectrum interference*. We measure continuously the pollutants and using our VANET communication we update the values onto a database via a web server as an interface, see Figure 8.14.

8.2.1 US EPA: Clean Air Act

The United States Environmental Protection Agency, EPA [119], is the institution responsible for observing and regulating the contaminant and polluting substances and their impact on human health. Enacted by Richard Nixon in 1970 via the executive order called Clean Air Act [121], the EPA has identified the following six major pollutants found in industrial areas of the United States:

1. Carbon Monoxide, CO
2. Ozone, O_3
3. Nitrogen Oxides, NO_x
4. Sulfur Oxides, SO_x
5. Lead, Pb
6. Particle Matter (1, 2.5 and 10µg/m^3)

From these pollutants, we decided to measure the following due to the sensor availability, price, and easiness to measure. The sensors are: *Carbon Monoxide (CO), Carbon Dioxide (CO_2), Ozone (O_3), Nitrogen Monoxide (NO), and Particle Matter (1, 2.5, and 10*µg/m^3*)*, see Figure 8.7 and 8.8.

8.2.2 MEX SEMARNAT: ProAire Mexicali

According to [111], in 1994 the federal government of México established certain norms and regulation to measure the pollution and its effects among the Mexican population. However, due to the lack of sufficient equipment and the rigorous studies, the federal government relied on the research and standards adopted internationally. As a consequence, the Norma Oficial Mexicana (NOM), or the federal official regulations for pollution that rule the quality of the air in México identifies the same six pollutants that the US EPA does in its Clean Air Act, Subsection 8.2.1.

The NOM that currently are in place in México are the following: NOM-020-SSA1-1993, NOM-021-SSA1-1993, NOM-022-SSA1-1993, NOM-023-SSA1-1993, NOM-024-SSA1-1993, NOM-025-SSA1-1993, NOM-026-SSA1-1993, which each ones corresponds to a particular gas mentioned in the Clean Air Act. For more information on these norms, we direct the reader to [111].

8.3 Testbed

For this project we proposed to study the following two elements natural to any Vehicular Network:

1. Communication
2. Pollution

A vehicular network, or VANET, is a network composed of highly mobile devices that collaborate in order to understand the environment where they are associated to and perform intelligent decisions regarding whatever purpose they take upon. One fundamental component of a VANET is the vehicles' communication; before any intelligence can be generated, the members of a VANET must be able to communicate. For this project, we addressed this issue with our study on the congestion/interference of the ISM band (Industrial, Scientific, and Medical) that commodity WiFi wireless cards use see Subsection 8.1. We created a test bed called *CogNet*, for Cognitive Radio Networks, that allows us to collect spectral information of the area surrounding a particular vehicle. The data collected allows us to have an idea of how vehicles, or any mobile station, should communicate. The CogNet test bed is described in section 8.3.1.

Another aspect particularly appropriate to a VANET is the collection of pollution data where the vehicles operate. As part of an addendum to our VANET proposal, we designed another test bed, an environmental test bed, that is capable of collecting pollution data harmful to humans, see Subsection 8.3.2. The environmental test bed is composed of Libelium™ sensors that are

Table 8.1: Libelium sensors

Sensor	**Description**[119]
CO (ppm)	Carbon Monoxide, a **criteria pollutant** gas
CO_2 (ppm)	Carbon Dioxide, a greenhouse gas
NO (ppm)	Nitrogen Monoxide, a **criteria pollutant** gas
O_3 (ppb)	Ozone, a **criteria pollutant** gas
PM 1μg/m^3	Particle Matter, a **criteria pollutant** particle
PM 2.5μg/m^3	Particle Matter, a **criteria pollutant** particle
PM 10μg/m^3	Particle Matter, a **criteria pollutant** particle
T (C)	Temperature
P (kPas)	Pressure
RH	Relative Humidity

Figure 8.1: Spectrum interference on the vehicles on the road

transmitted over a wireless/cellular network to our databases at UCLA. This data is available online via a web page for anyone to see in order to perform future environmental studies, see Figures 8.15 and 8.16. The description of the environmental test bed is fully described in Section 8.3.2, and the database design is discussed in Section 8.3.3.

8.3.1 CogNet – The Cognitive Radio Network

CogNet, or Cognitive Radio Network, is a network capable of adapting to the conditions of the electromagnetic spectrum, see Section 8.1.

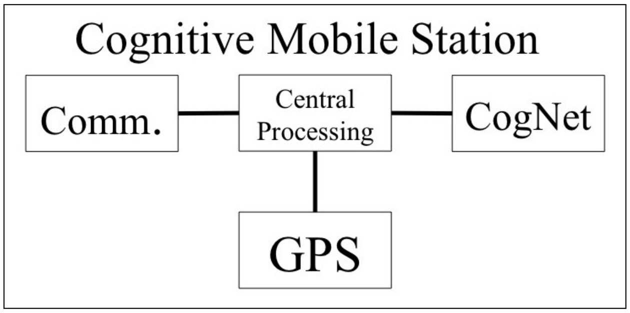


Figure 8.2: CogNet Architecture of a MS

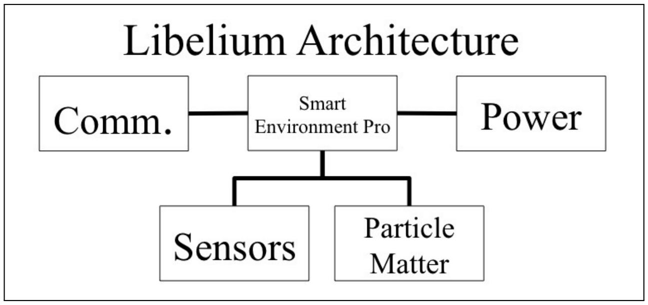


Figure 8.3: The Libelium™Architecture

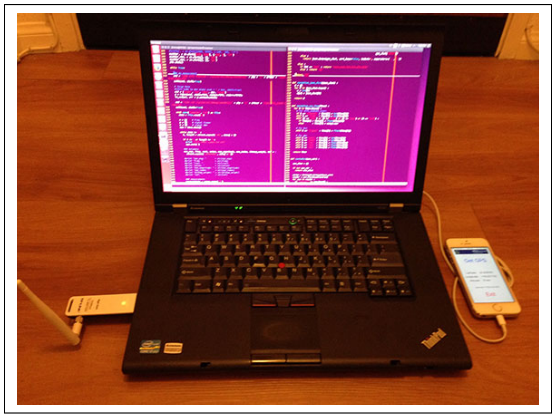


Figure 8.4: CogNet Mobile Station

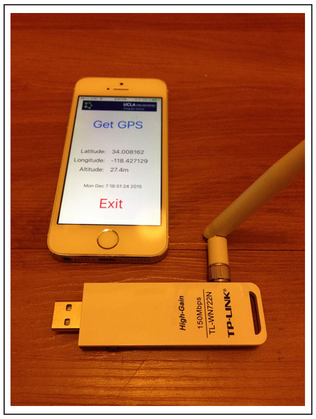
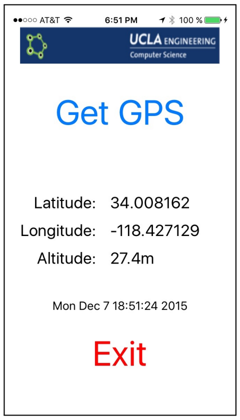


Figure 8.5: USB Dongle and iPhone as GPS



Latitude: 34.008162
Longitude: -118.427129
Altitude: 27.4m

Mon Dec 7 18:51:24 2015

Figure 8.6: iOS GPS app and USB Dongle

Our CogNet is composed of several cognitive capable *Mobile Stations* following the diagram shown in Figure 8.2. A mobile station is a commodity PC laptop with the Ubuntu 14.04 LTS operating system, equipped with at least two IEEE 802.11 wireless interfaces for network connectivity; see Figures 8.4, 8.5, and 8.6. One of the wireless interfaces is dedicated for communications and establishes connectivity to a VANET while the other is dedicated for spectrum scanning and it is what makes the mobile station cognitive capable.

The main unit in charge of all the processing of the data received by the auxiliary units is the *Central Processing* in Figure 8.2. This module is a group of programs in charge of the processing of the feedback provided by the CogNet unit and the GPS unit, Figure 8.5. It also reconfigures the Communication unit in order to adapt to the spectrum characteristics around the mobile station.

The module in charge of the cognition in the mobile station is called the *CogNet* unit. This unit is composed of one IEEE 802.11 Atheros™cards with its driver loaded onto the Linux kernel in debug mode. Under debug mode, the Atheros™driver makes available to the root user of the operating system several collected statistics, including the spectrum readings after executing a wireless scan. We use a commodity USB dongle that supports the specified driver to add this dedicated interface onto the equipment, see Figure 8.5. Since scanning the entire 2.4GHz ISM band and 5GHz U-NII takes some time, we managed to bring it down to an average of $3.5 seconds$ per scan; therefore, the cognitive module provides with spectrum feedback to the Central Processing module every so much time.

When the CogNet unit performs an environmental scan, we also keep the location where this scan took place as well as the time; in other words, we geotag the scan data using the *GPS* unit. This unit is composed of a mobile phone that runs an cellular application capable of taking responding with a JSON format geotag upon demand. The phone application is shown in Figure 8.5 and it was developed at our laboratory in UCLA.

8.3.2 Libelium™– The sensor test bed

The sensor test bed, or Libelium™, is the equipment that measures the environment for pollutants; the architecture is shown in Figure 8.3. The main board is the *Smart Environment Pro*. It is

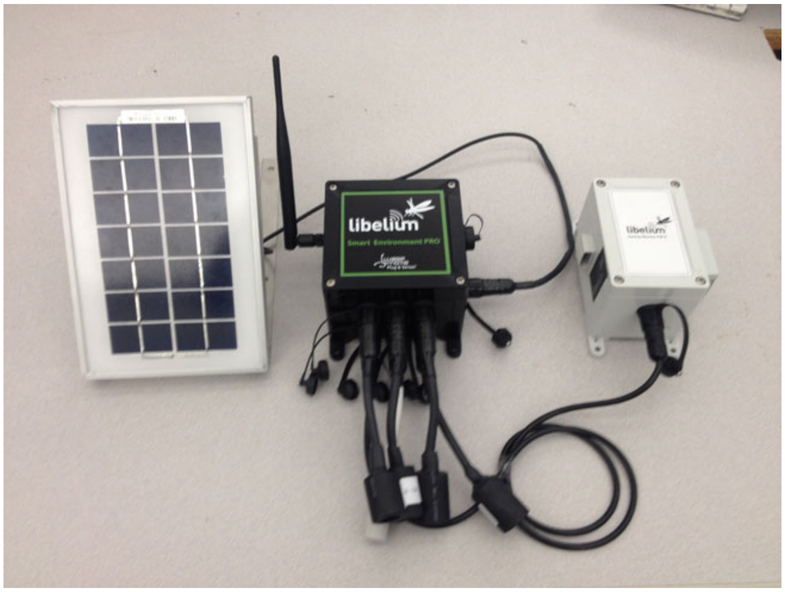


Figure 8.7: Libelium™Smart Environmental Pro unit with wireless transceiver and solar power supply

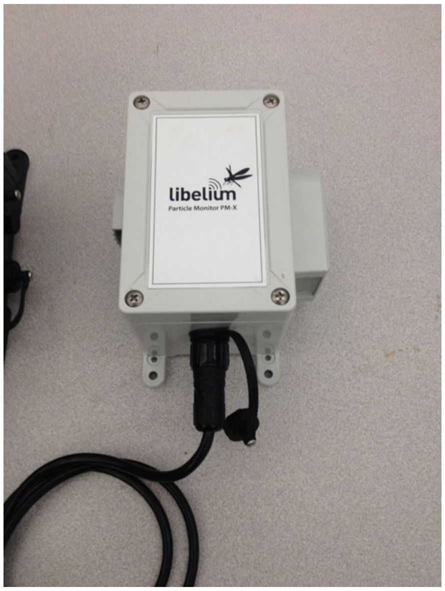


Figure 8.8: Particle Matter unit

in charge of the sensor and wireless communication configuration, the periodic sampling of the sensors, and the packetizing of the sensorial data that will be delivered to the UCLA database via the cellular interface (the iPhone App), see Figures 8.11, 8.12, and 8.13.

The *Sensors* unit is made up of a group of up to six sensors that may be plugged into the main board. Each sensor works independently and may be sampled by the main board in predefined intervals of time once the sensor has been "warmed up." Each sensor may have different sampling intervals, so the main board has to take this value into consideration when sampling all the sensors at once. Currently we have the following sensors installed onto a Libelium™Smart Environment Pro main board: Carbon Monoxide (CO), Nitrogen Monoxide (NO), Carbon Dioxide (CO2), Ozone (03), Temperature, Pressure, and Relative Humidity (all three in one sensor), and the Particle Matter unit that has the following three sensors: Particle Matter 1, 2.5, and $10\ \mu\mathrm{g}/\mathrm{m}^3$. The sensors can be seen in Figure 8.7.

The *Particle Matter* unit, as described before, is shown in Figure 8.8. It is in charge of counting how many particles of three distinct sizes are there in a cubic meter of volume. The particle sizes are 1, 2.5 and $10\ \mu\mathrm{g}/\mathrm{m}^3$; each has a distinct role in the environmental pollution and health hazard. This unit is encased in its own frame since the measurement of particles requires room to perform

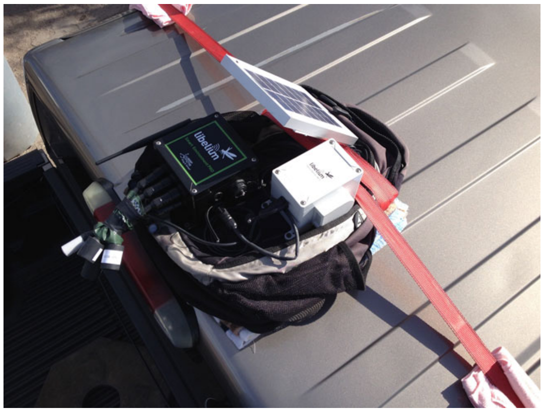


Figure 8.9: Libelium Testbed

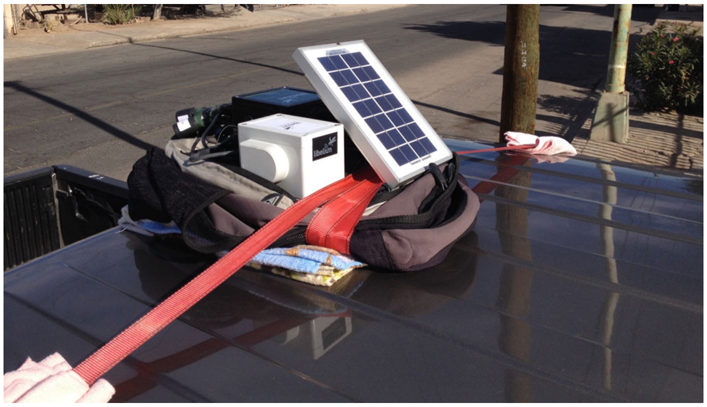
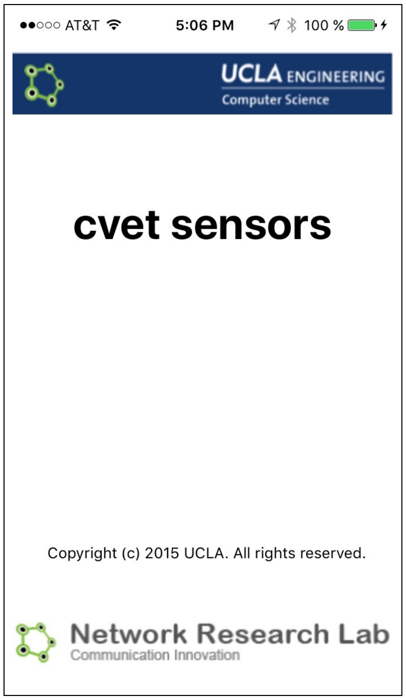


Figure 8.10: Libelium Testbed

its particle counting. Moreover, the inflow of air from the environment must be constant and stable; in other words, air must flow in and out. The unit is equipped with a side fan that pulls air from the outside environment; this posed a problem to us when we installed the unit on a moving vehicle: the rate of air coming in and out of a moving is not constant. We placed the unit backward, opposing the incoming flux of air and use the case protecting the electronics of the Particle Matter sensor as barriers. This way, the fan is able to sample the air that is going in the opposite direction of the moving vehicle at a much more stable rate, see Figure 8.9 and 8.10 for the mounting setup.

The *Power* unit is the major source of power of the device. It is composed of a solar panel and a battery pack; depending of the duty cycle of the sensor test bed, it keep the machine active for up to five hours. At night, when the test bed does not receive sun power, the test bed uptime reduces to about one hour. The reason for this is because the Smart Environment Pro was designed to have duty cycles of four to five minutes; that sampling period is not acceptable for a moving vehicle. Our sampling time is in the lower tens of seconds (10 seconds in most of the cases). This stress of work imposed onto the sensors consumes a large amount of power resources. While a battery pack designed to be autonomous (solar powered plus battery), in our case, the entire power unit is not able to maintain the system running for more than a few hours without turning it off and let it recharge. This situation is fine for us now since we perform measurements by driving a vehicle for no more than three hours. However, longer sampling requirements would need another source of power or lower duty cycle time. The power unit is installed together with the main unit and it may be seen in Figure 8.7.



cvet sensors

Copyright (c) 2015 UCLA. All rights reserved.

Figure 8.11: iOS cvet-app front page

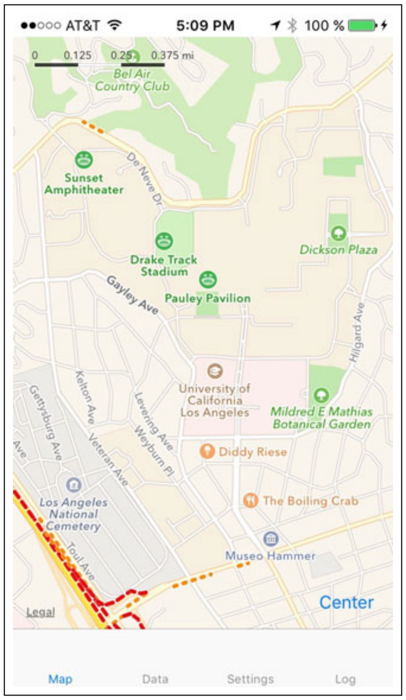
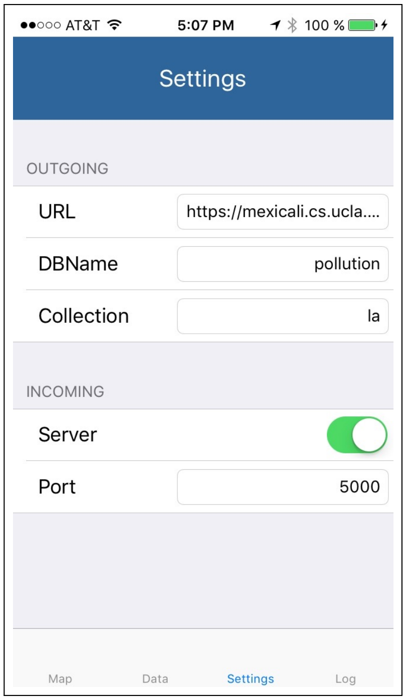


Figure 8.12: CVeT Map View



Settings

OUTGOING

URL https://mexicali.cs.ucla....

DBName pollution

Collection la

INCOMING

Server ●

Port 5000

Figure 8.13: CVeT Settings View

The *Communication* unit is in charge of transferring the data from the mobile environment to the UCLA database via the Internet. It is composed of a standard IEEE 802.11 WiFi wireless card that connects to a mobile hot spot created by an autonomous Apple iPhone device (not physically dependent of the test bed). We could have opted for installing a GPRS interface (a cellular uplink) on the main board but we deemed it an unnecessary expense. Once the main board produces a packet of data that needs to be delivered, it uses the WiFi link to deliver it to a mobile application running on the iPhone. We also develop an in-house iPhone iOS application to control the flow of data; screenshots of the app are in Figures 8.11, 8.12, 8.13.

When the iPhone app receives a message from the main board wirelessly, it decodes the message packetized as a JSON document. The app is able to extract the current sensorial data and tags it with a time stamp and a geographical coordinates (a geo-temporal tag). Immediately next, the app displays the dot onto the Map view, Figure 8.12, stores it locally on the Data view logs the event on the Log view and delivers it to the database at the UCLA campus. This procedure is done at every sampling event received by the iPhone app with is approximate the same as the duty cycle time configured in the main board.

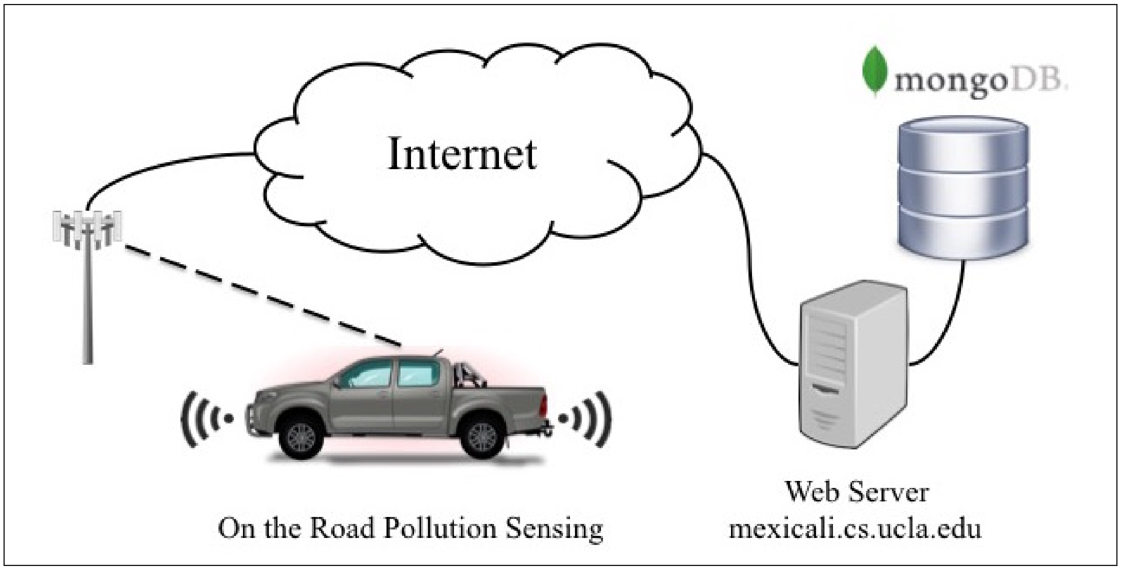


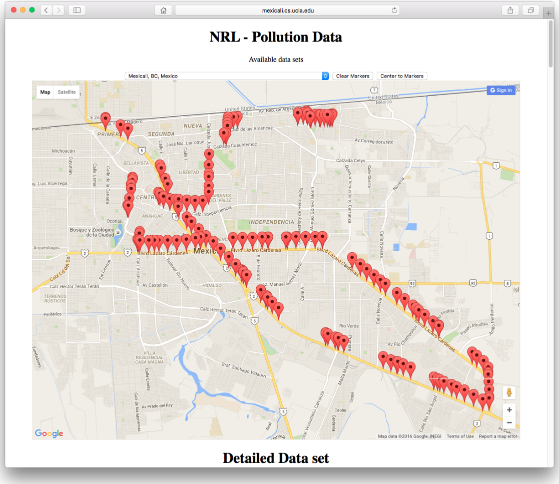
Figure 8.14: Data collection and storage

8.3.3 MongoDB™– The database and web services

The back-end *database*, or the *mongo* database, is the location where we store all the sensorial data collected on the road. It feeds with the data delivered by the iPhone application and stores it in a group of tables. Unlike regular databases that store records of data, MongoDB™is a database that stores JSON documents. A *JSON Document* is a text string that follows a well-defined format described here [70]. It is especially beneficial for data delivery on mobile devices due to the simplicity of its format and the light-weight in overhead (it produces short packets), as opposed to XML formatted data.

When the libelium test bed produces one message of data, it is delivered to the iPhone app for further processing and geo-temporal tagging, Figure 8.11. Once the sensor data sample is ready, it is converted onto a JSON document and it is delivered to a Web Server, or the *server* via a POST RESTful request. The server acts as the middle man between the mobile collection devices and the database and it is in charge of dissecting the received JSON document in order to decide on which *collection* this new *sample* must be stored, see Figure 8.14.

In MongoDB, a collection is the equivalent of a table on a regular DBMS, however, it is called a collection because it stores a collection of JSON documents. A sample is the sensorial data that the Libelium test bed produces and geo-temporal tagged by the iPhone app. In this manner,

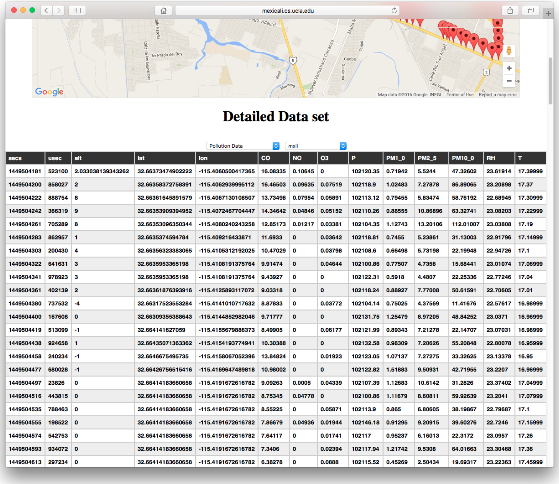


NRL - Pollution Data

Available data sets

Detailed Data set

Figure 8.15: Map view



Detailed Data set

secs	usec	alt	lat	lon	CO	NO	O3	P	PM1_0	PM2_5	PM10_0	RH	T
1449504181	523100	2.033038139343262	32.66373474902222	-115.4060500417365	16.08335	0.10645	0	102120.35	0.71942	5.5244	47.32602	23.61914	17.39999
1449504200	858027	2	32.66358372758391	-115.4062939995112	16.46503	0.09635	0.07519	102118.9	1.02483	7.27878	86.89065	23.20898	17.37
1449504222	888754	8	32.66361645891579	-115.4067130108507	13.73498	0.07954	0.05891	102113.12	0.79455	5.83474	58.76192	22.68945	17.30999
1449504242	366319	9	32.66353909394952	-115.4072467704447	14.34642	0.04846	0.05152	102110.26	0.88555	10.86896	63.32741	23.08203	17.22999
1449504261	705289	8	32.66353096350344	-115.4080240243258	12.85173	0.01217	0.03381	102104.35	1.12743	13.20106	112.01007	23.03808	17.19
1449504283	862957	1	32.6635374594784	-115.409216433871	11.6933	0	0.03642	102118.81	0.7455	5.23861	31.13003	22.91796	17.14999
1449504303	200430	4	32.66356323383065	-115.4105312192025	10.47029	0	0.03798	102108.6	0.66498	5.73198	22.19948	22.94726	17.1
1449504322	641631	3	32.6635953365198	-115.4108191375764	9.91474	0	0.04644	102100.86	0.77507	4.7356	15.68441	23.01074	17.06999
1449504341	978923	3	32.6635953365198	-115.4108191375764	9.43927	0	0	102122.31	0.5918	4.4807	22.25336	22.77246	17.04
1449504361	402139	2	32.66361876393916	-115.4125893117072	9.03318	0	0	102118.24	0.88927	7.77008	50.61591	22.70605	17.01
1449504380	737532	-4	32.66317523553284	-115.4141010717632	8.87833	0	0.03772	102104.14	0.75025	4.37569	11.41676	22.57617	16.98999
1449504400	167608	0	32.66309355388643	-115.4144852982046	9.71777	0	0	102131.75	1.25479	8.97205	48.84252	23.0371	16.96999
1449504419	513099	-1	32.664141627059	-115.4155679886373	8.49905	0	0.06177	102121.99	0.89343	7.21278	22.14707	23.07031	16.98999
1449504438	924658	1	32.66435071363362	-115.4154193774941	10.30388	0	0	102132.58	0.98309	7.20626	55.20848	22.80078	16.95999
1449504458	240234	-1	32.6646675495735	-115.4158067052396	13.84824	0	0.01923	102123.05	1.07137	7.27275	33.32625	23.13378	16.95
1449504477	680028	-1	32.66426756515416	-115.4169647489818	10.98002	0	0	102122.82	1.51883	9.50931	42.71955	23.2207	16.96999
1449504497	23826	0	32.66414183660658	-115.4191672616782	9.09263	0.0005	0.04339	102107.39	1.12683	10.6142	31.2826	23.37402	17.04999
1449504516	443815	0	32.66414183660658	-115.4191672616782	8.75345	0.04778	0	102100.86	1.11679	8.60811	59.92639	23.2041	17.07999
1449504535	788463	0	32.66414183660658	-115.4191672616782	8.55225	0	0.05871	102113.9	0.865	6.80605	38.19867	22.79687	17.1
1449504555	198522	0	32.66414183660658	-115.4191672616782	7.86679	0.04936	0.01944	102146.18	0.91295	9.20915	39.60276	22.7246	17.15999
1449504574	542753	0	32.66414183660658	-115.4191672616782	7.64117	0	0.01741	102117	0.95237	6.16013	22.3172	23.0957	17.26
1449504593	934072	0	32.66414183660658	-115.4191672616782	7.3406	0	0.02394	102117.94	1.21742	9.5308	64.01663	23.30468	17.36
1449504613	297234	0	32.66414183660658	-115.4191672616782	6.38278	0	0.0888	102115.52	0.45269	2.50434	19.69317	23.22363	17.45999

Figure 8.16: Table view

sensorial data samples are created at the Libelium test bed, geo-tagged and delivered at the iPhone app, received and dissected at the Web Server, and stored at the mongo database.

8.4 Measurements

In this section we are going to show the experiment results collected with our testbeds described in Subsections 8.3.1 and 8.3.2 on the road.

8.4.1 CogNet Measurements

We first describe the methodology of our experiments and then we show the results obtained.

8.4.1.1 Methodology

In this section we describe our experiment setup and discuss the methodology used. The spectrum measurements were performed during the third week of November, just before the Thanksgiving break at UCLA. This selection assumes that most of the students are concentrated our around the UCLA campus or the Westwood University village, in places like the coffee shops, restaurants, cinemas, or grocery markets. Moreover, another set of experiments were done on a vehicle to

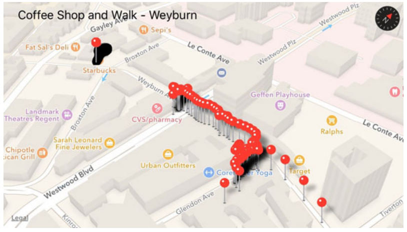


Figure 8.17: Weyburn Ave.

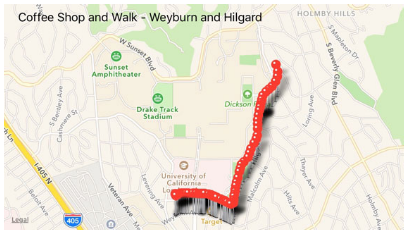


Figure 8.18: Highard Ave.

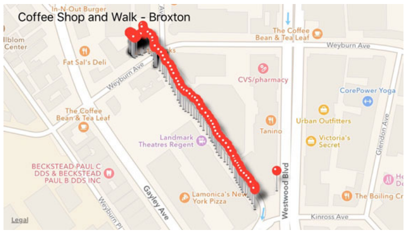


Figure 8.19: Broxton Ave.

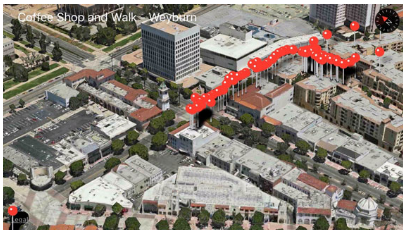


Figure 8.20: Weyburn Ave.

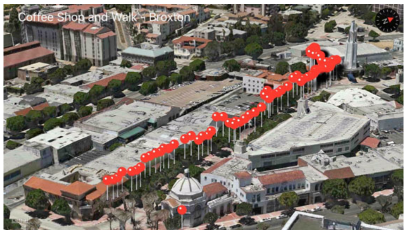


Figure 8.21: Broxton Ave.

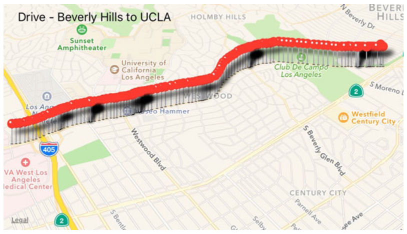


Figure 8.22: Wilshire Blvd.

measure the spectrum interference onto a VANET. The vehicle main trajectory is the Wilshire Blvd, a very busy artery of West Los Angeles, from the I-405 freeway through the end of Beverly Hills, CA, in both eastbound and westbound directions; Figure 8.22 shows the trajectory on a map and Figures 8.23 and 8.24 show the flyover view of the Wilshire corridor. The average vehicle speed is $35\ mph$ but the traffic at the time resembles a bumper-to-bumper condition, so there are situations in our measurements that the vehicle is stopped for several seconds.

For the walking experiments, we took one cognet mobile station, equipped with one USB dongle and one iPhone device tagging each spectrum sample taken with a geographical, temporal timestamp. We setup three distinct scenarios described next.

Coffee Shop and Walk – Weyburn During this scenario, the subject stayed half of the time at a local coffee shop on the corner of Weyburn Ave. and Broxton Ave. in the Westwood village. Next, the subject proceed to leave the coffee shop and walk along Weyburn Ave. crossing Westwood Blvd. and made a right onto Glendon Ave. to stop at another popular restaurant/coffee stop of the university village; see Figure 8.17. On this trajectory the subject encountered several spots with high and low spectrum interference. For example, there are three coffee shops, two movie theaters, at least five restaurants, a pharmacy, two grocery stores, and a major retailer store. Figure 8.20 shows the flyover view of this trajectory. Also, there are several spots of no interference; there are

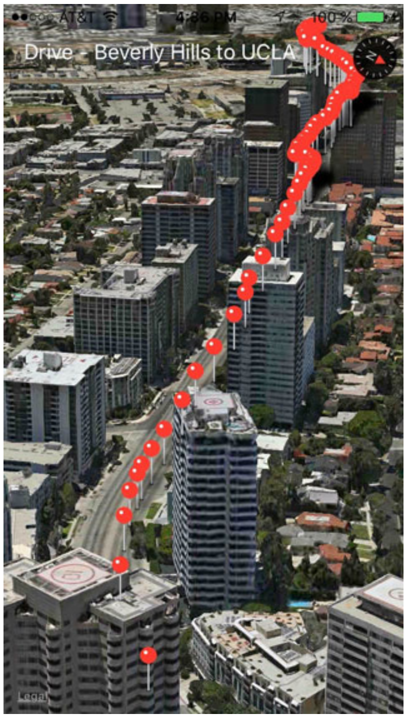


Figure 8.23: Wilshire Blvd at Westwood

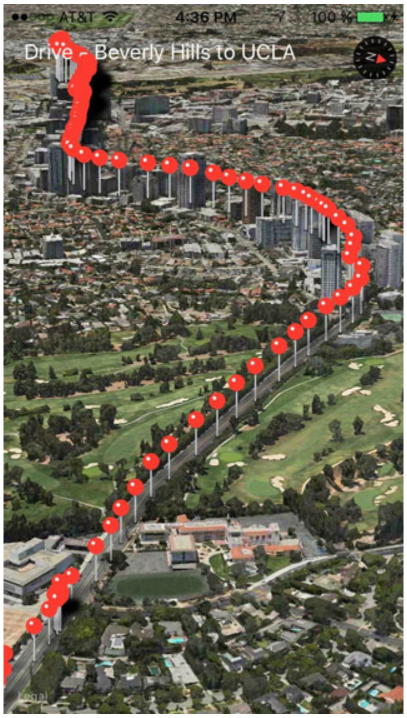


Figure 8.24: Wilshire Blvd at LA Country Club

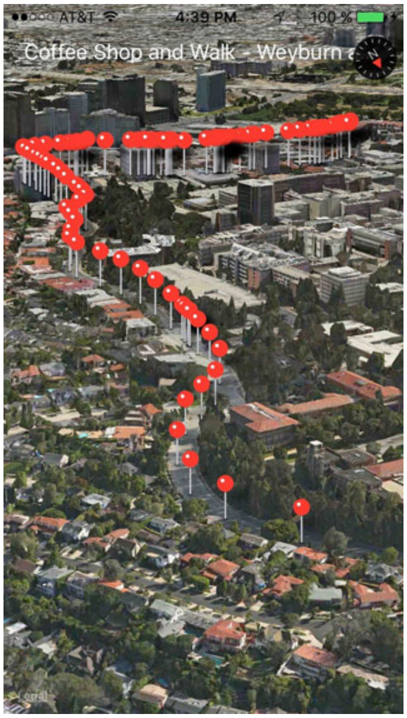


Figure 8.25: Higard Ave. at UCLA East

areas with high walls or no entrance doors where the amount of people is limited or there is no people using the Internet on their mobile devices. This experiment is performed by walking and carrying the antennae of the cognitive interface outside a backpack and clear in the open.

Coffee Shop and Walk – Weyburn and Higard This scenario represents a long walk from the same coffee shop as in the previous scenario but this time it ends at a well-known bus stop on the east side of the UCLA campus, see Figure 8.18. This scenario studies the interference on the road that a pedestrian or a bicyclist may find while using the streets. This trajectory is composed of a busy commercial center and a populated residential area where students and university neighbors dwell; see Figure 8.25 for a flyover view. The walk was performed at a stead rate without major stops, except those at intersections obeying the road signs. It starts at the corner of Weyburn Ave. and Broxton Ave. and proceeds eastward onto Weyburn Ave. crossing Westwood Blvd, and the Glendon and Tiverton Avenues before the subject turned left and northward on Highard Ave. Finally, the only major intersection stop is on Le Conte Ave. before the subject stops at the UCLA Bus Stop.

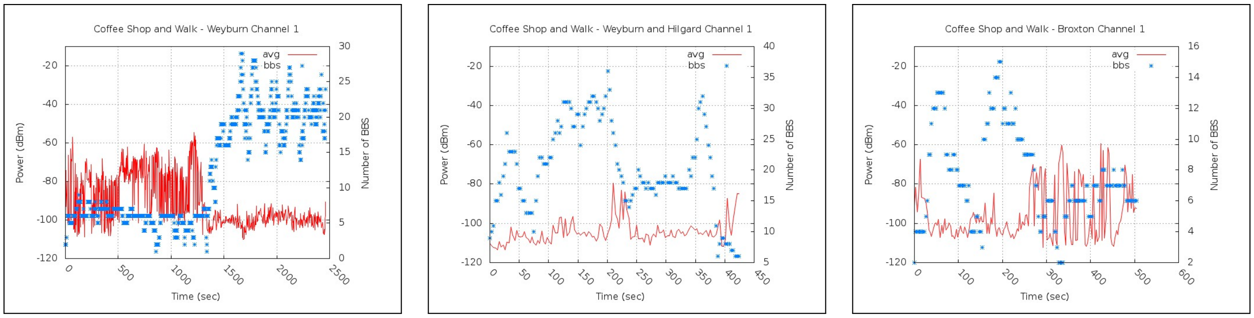
Coffee Shop and Walk – Broxton This scenario studies the interference present only at a busy commercial street; see Figure 8.19. The chosen street is Broxton Ave. and it has several restaurants where students and visitors spend most of their free time lunching or dining; see the flyover view in Figure 8.20. The subject starts walking at the corner of Weyburn and Broxton Avenues and proceeds southbound on Broxton Ave. until the next intersection on Broxton Ave. and Kinross Ave. As opposed to the second scenario above, this scenario is shorter and more centered towards representing the interference in a non-residential area.

Drive – UCLA to Beverly Hills This scenario represents the first of three driving measurements taken along the Wilshire Blvd. in Los Angeles, CA; see Figure 8.22. As explained before, the average speed of the vehicle is 35 mph but these experiments were purposely taken during the rush hours on the evening. This makes sense because when a vehicle moves at a high speed, the interference that may exist at a particular location becomes irrelevant just a few seconds later. Moreover, vehicle devices such as travel planners (GPS, etc.) and mobile devices may become active with higher probability. Specifically, this trajectory is composed of a lots of high tower buildings, mostly residential, that can be seen in Figure 8.23 and later followed by an empty zone, the Los Angeles Country Club in Figure 8.24, where we expected to find little to no interference.

Drive – Beverly Hills Westbound and Eastbound This scenario begins at the corner of Wilshire Blvd. and Santa Monica Blvd., this is the beginning of the City of Beverly Hills, CA, and it continues eastbound until the other end of Beverly Hills, at the corner of Wilshire Blvd. and San Vicente Blvd. This trajectory covers the business district of Beverly Hills, especially the first half or the western side of the trajectory. It is composed of many stores, boutiques, hotels, restaurants, banks, pharmacies, a movie theater, etc. It aims to study the interference at a business district on a moderate speeding vehicle.

8.4.1.2 Measurements

We start the spectrum measurements with the first scenario described in section 8.4.1.1. Figure 8.26a is the longest measurement that we took because it combines approximately one hour and



(a) Coffee Shop and Commercial zone

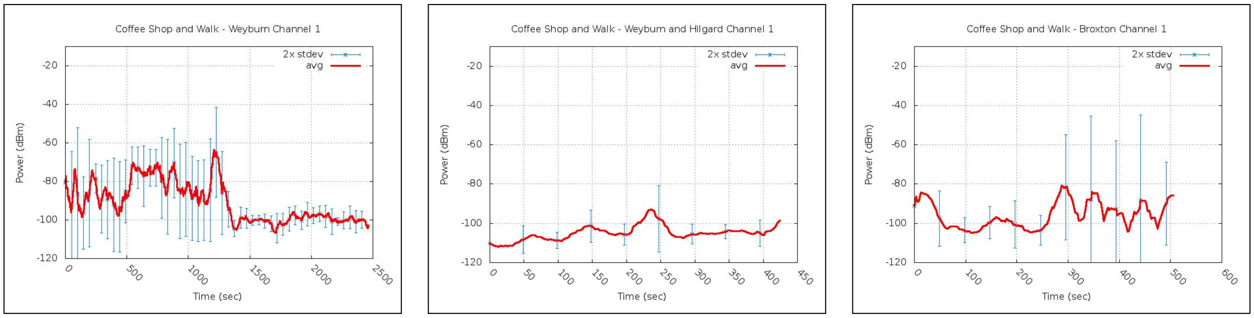
(b) Commercial and Residential area

(c) Commercial area over a narrow street

Figure 8.26: Interference measurements and number of access points detected on our walking experiments.

fifteen minutes of spectrum measurements at a coffee shop, where a large number of students are present working on their computers and mobile devices. As the figure shows from the period range $[0, 1700]$, it is clear that the spectrum occupancy is large, with most of the measurements averaging around the $-80dbm$ of power present on channel 1. Also notice the low amount of base stations, (BBS), present. Inside a crowded room, within the same period we can observe that the number of hot spots averages 5. This is expected since the hot spot that serves the coffee shop may be advertising its own network and the others may come from neighboring restaurants. Notice the contrast in both the number of base stations and interference when the subject left the coffee shop at the approximate time $t = 1700$ and started a walk on Weyburn Ave. We see a significant increase in the number of hot spots detected while the interference decreases abruptly. This shows that a when a point of measurement is static next to a high congestion zone like a hotspot, it will find a large amount of interference concentrated on a few number of hot spots but while on the move, the interference on the road decreases and a more stable spectrum is observed.

Similar results are shown in Figures 8.26b and 8.26c. The subject in 8.26b walked through two zones: a commercial and a residential zone. We observe that the interference is not very significant while on the road; however, we are able to detect the number of hot spots along the walk. These hot spots are clearly from the commercial shops and the student apartments in the area. Notice, though, on Figure 8.26c that it is still possible to be interfered on the road during a walk by a group of people actively using their mobile stations. This is shown in the time range $[270, 500]$ of Figure 8.26c.



(a) Coffee Shop and Commercial zone

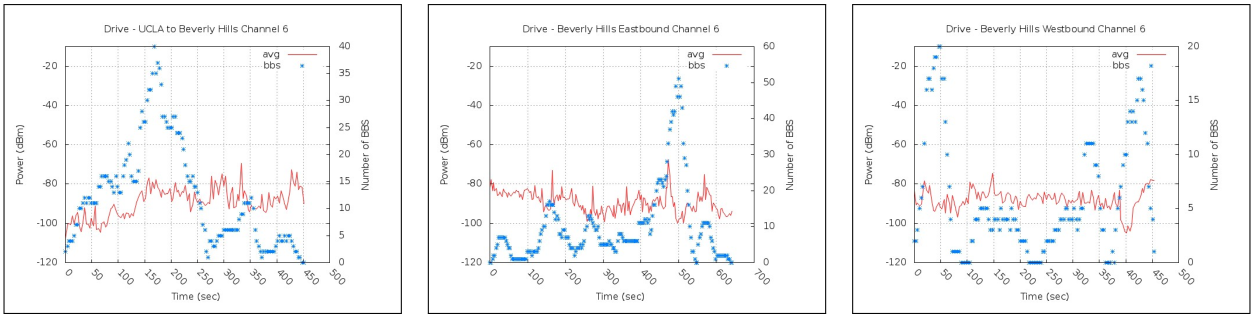
(b) Commercial and Residential area

(c) Commercial area over a narrow street

Figure 8.27: Moving Window Average and two standard deviation (99% of the samples) of Figure 8.26.

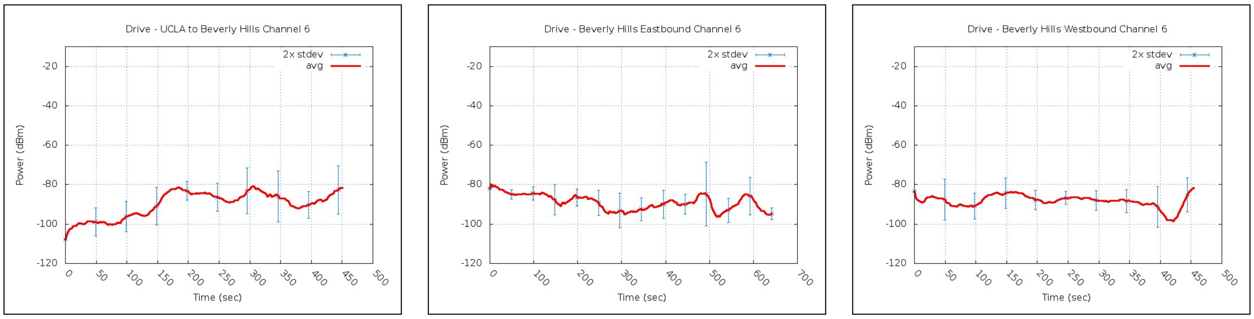
One of our goals for the spectrum analysis in a CogNet is to determine when to use a channel and when to switch onto another channel. On this goal, we decided to calculate the moving window average, with a window $wnd = 50$, and plotted the standard deviation within this window. The plots for the three scenarios shown in Figures 8.27a, 8.27b, and 8.27c. Notice how standard deviation clearly identifies the zone of spectrum occupancy in Figure 8.27a and Figure 8.27c. We plotted the value $2 \times stdev$ because it represents a little more than 95% of the data. We conclude that this measurement clearly distinguish if the spectrum on a particular channel is free or busy.

On our measurements performed on the road we found that the spectrum is much more stable than on our walking experiments. Figures 8.28a, 8.28b, and 8.28c sustain our claims. Notice how on these figures the interference dwells mostly below the $-80dbm$, which is indicates power present on the channel but very far away. In terms of the number of hot spots detected, notice the high peaks on the three figures. On Figure 8.28a we see a high peak in the time range $[50, 250]$; this corresponds to the Wilshire Blvd. corridor where most of the high residential towers are. On Figure 8.28b we see the same peak in $[400, 550]$, this is the area in Beverly Hills, the east, where there are many houses. It is clear that while on the road, on a moving vehicle at an average speed of 35 mph, the commercial and residential interference is almost negligible, even though the local hot spots are detected by the moving car. Figures 8.29a, 8.29b, and 8.29c support that claim. In all three we see the standard deviation measurement very stable throughput the entire collection process.



(a) Wilshire Blvd. Westbound (b) Beverly Hills, Westbound (c) Beverly Hills, Eastbound

Figure 8.28: Interference measurements and number of access points detected on a drive from Wilshire Blvd. at I405 through Beverly Hills, CA.



(a) Wilshire Blvd. Westbound (b) Beverly Hills, Westbound (c) Beverly Hills, Eastbound

Figure 8.29: Moving Window Average and two standard deviation (99% of the samples) of Figure 8.28.

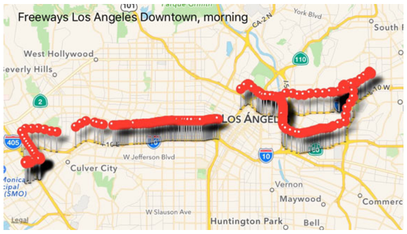


Figure 8.30: I-10, morning

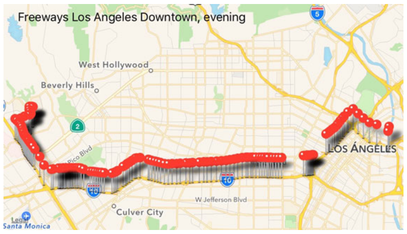


Figure 8.31: I-10, evening

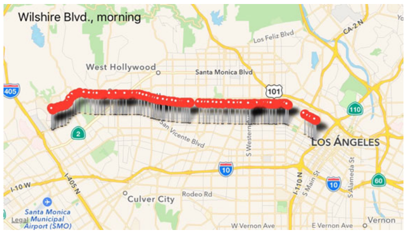


Figure 8.32: Wilshire Blvd.

8.4.2 Pollution Measurements

In this section we describe the experiment results collected with our mobile environmental testbed deployed on the roads of the cities of Los Angeles, CA, Moreno Valley, CA, and Coachella Valley, CA, all in the United States, and on the city of Mexicali, BC, Mexico.

8.4.2.1 Methodology

As described above, we performed our measurements on the road, with the libelium testbed attached onto the roof top of a pick up truck, as seen in Figures 8.9 and 8.10. and collected pollution samples of the area. On all these experiments, the speed of the vehicle went from 40 to $70mph$; at the lower speeds, our vehicle sampled major streets on several cities while at the higher speeds the vehicle sampled the environment while driving on a freeway or a highway. Next we describe the scenarios where the sampling took place.

Los Angeles Area The region of Los Angeles that we scanned is composed by several freeways (i10, CA101, and CA110) on both directions, eastbound and westbound, and a major artery (Wilshire Blvd. from UCLA to downtown LA). We recorded the geographical locations of our trajectories; Figure 8.30 shows the trajectory that we took from west Los Angeles towards Los Angles downtown via the freeway i10 during the morning at around 7am. We chose this time because it is right at the rush-hour traffic on the major freeways. We traveled eastwards and then we drove around the Los Angeles downtown freeway system in order to sample as much as the LA area as possible. Figure 8.31 shows the traffic on the same trajectory but this time westbound from Los Angeles downtown back to the UCLA campus during the evening at around 5pm, again, during

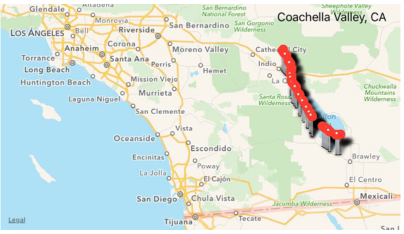


Figure 8.33: Coachella Valley

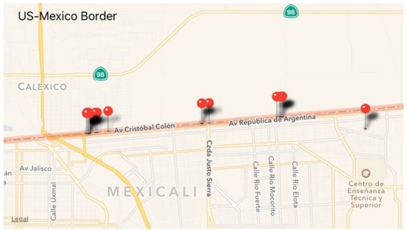


Figure 8.34: US-Mex Boarder

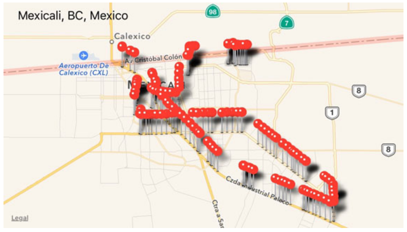


Figure 8.35: Mexicali, BC

the rush-hour period.

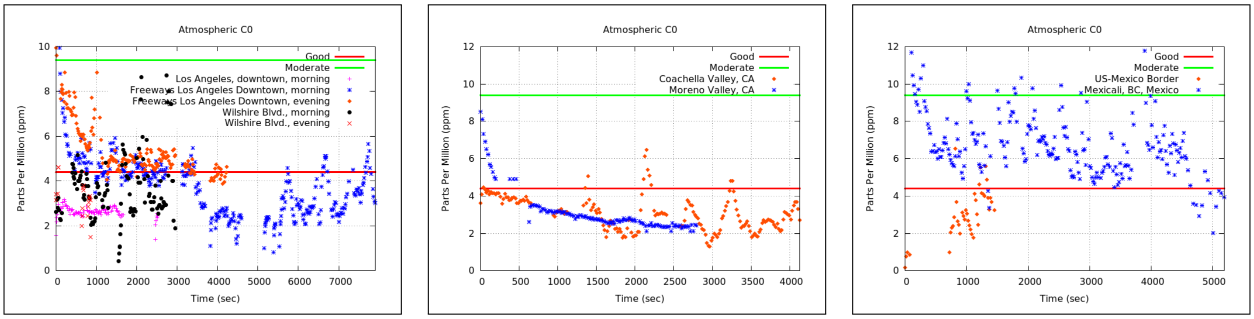
Inland Empire The Inland Empire region includes the cities of Moreno Valley, CA and Coachella Valley, CA. The city of Moreno Valley, CA, is a city next to Riverside, CA, in the California county of Riverside. It is a city with little industrial facilities, mostly service industries that experiences little to no air pollution. Located in the northern mountains of Riverside, it is approximately $500m$ above the sea level and $160m$ above Riverside; this means that the major air contaminants stay at Riverside as the mountains provide a barrier for it. Both the cities of Moreno Valley and Coachella serve as examples of clean air urban areas. We did not drive around this city, we only took measurements at a fixed location.

Coachella Valley, CA is a major agricultural area of the Inland Empire. Figure 8.33 shows the trajectory that we took. We started our measurements at the city of Coachella Valley, CA, on the California highway 86 driving southwards during the evening, 9pm approximately. The amount of vehicles on the road were minimal (1 or 2 vehicles per minute), so this scenario represents another clean area on a highway. It contrast our measurements from our previous scenario where we drove in rush-hour traffic.

Mexicali, BC, México Our last scenario includes the city of Mexicali, Baja California, México. Mexicali is a major industrial town with known air pollution problems and harsh temperatures at the center of the city. This combines with major agricultural activities in the south, which produces large amount of suspended particles. However, the major characteristic of Mexicali in terms of pollution is the amount of air particles floating on the air. Those particles are mostly dust and uncombusted fuel. Figure 8.35 shows the trajectory that we drove while taking our measurements

Table 8.2: Air Quality Index

Category	**CO (ppm)**	**O_3(ppb)**	**PM 1, 2.5** µg/m^3	**PM 10** µg/m^3
Good	4.4	54	12.0	54
Moderate	9.4	70	35.4	154
Unhealthy for Sensitive Groups	12.4	85	55.4	254
Unhealthy	15.4	105	150.5	354
Very Unhealthy	30.4	200	250.4	424
Hazardous	30.4+	200+	250.4+	424+



(a) Los Angeles, CA

(b) Coachella Valley, CA

(c) Mexicali, BC

Figure 8.36: CO measured in ppm

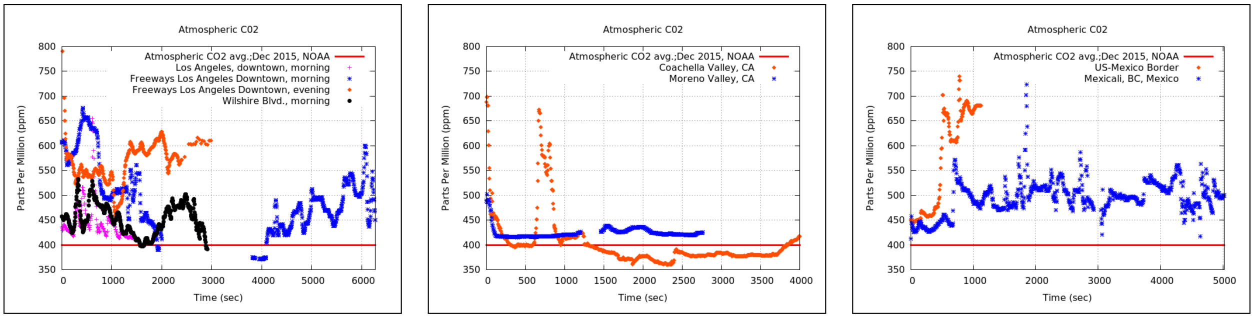
during the morning at approximately 8am. We drove major streets of the city as well as visited the southern side, where the agriculture areas start.

Air Quality Index The air quality index categorizes the hazard level of the concentration of a particular pollutant in the atmosphere. The following table shows level breakpoints of pollution concentration in the atmosphere and their respective category according to For the CO_2, we used the value announced on the page.

8.4.2.2 Measurements

We now present our findings.

Figure 8.36a shows the concentration levels of CO in the city of Los Angeles, 8.4.2.1. The



(a) Los Angeles, CA

(b) Coachella Valley, CA

(c) Mexicali, BC

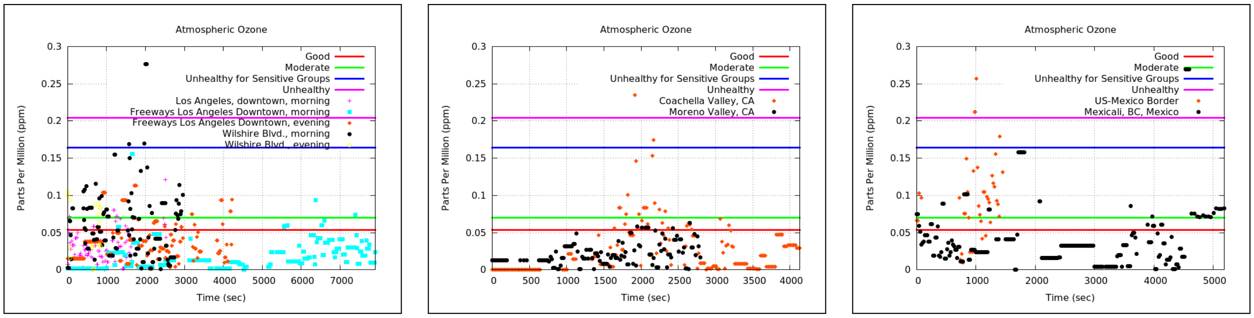
Figure 8.37: CO_2measured in ppm

figure shows that the amount of concentration of the pollutant gas are between the good and moderate levels. Notice the values around the time range $[0, 400]$; this area corresponds to morning trip eastbound towards downtown Los Angeles. Also for the measurements on the evening, the same time range corresponds to the pollution in downtown Los Angeles, westwards. The second time range, $[5000, < 8000]$, only contains values for the westwards trip towards downtown Los Angeles, and we can see that the levels are mostly in the good category. From this graph we can conclude that the amount of CO at or around Los Angeles under severe traffic conditions are acceptable.

In the second region, the Inland Empire 8.4.2.1, has much better concentrations of CO particles. This zone has little vehicular activity and it is either protected from air pollution from natural barriers (Moreno Valley) or it is far from major urbanization (Coachella Valley). Almost all the measurements are below the good category limits; therefore, we can conclude that these areas have very acceptable CO levels.

Lastly, the Mexicali, 8.4.2.1, has a concentration of CO that is mostly in the moderate category range. These readings are similar to those found in Los Angeles are, however, the amount of vehicles in Mexicali is largely inferior. We suspect that the excess of CO that we are observing in our figure that is not generated by the burning of vehicle fuel must be from random trash and urban fires.

We continue now with our measurements of CO_2. Figures 8.37a and 8.37c show the amount of carbon dioxide present in the environment; these figures correspond to major urban areas, Los Angeles and Mexicali, respectively. The amount of vehicles and population are considerably larger

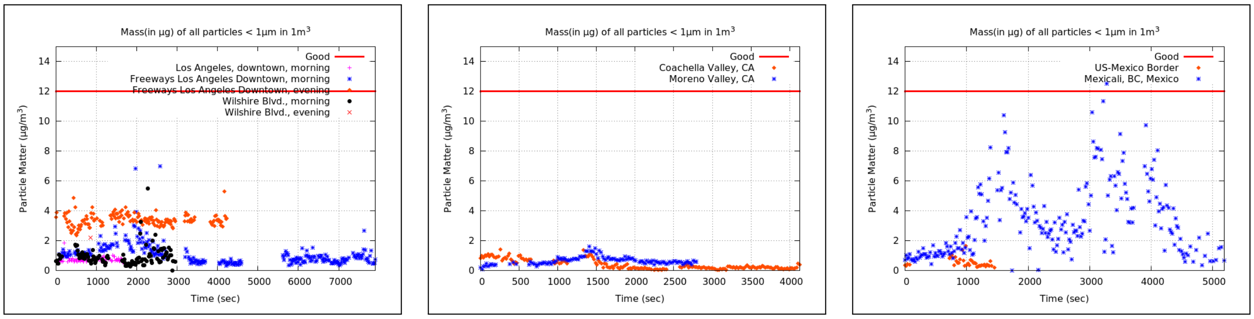


(a) Los Angeles, CA

(b) Coachella Valley, CA

(c) Mexicali, BC

Figure 8.38: O_3measured in ppm



(a) Los Angeles, CA

(b) Coachella Valley, CA

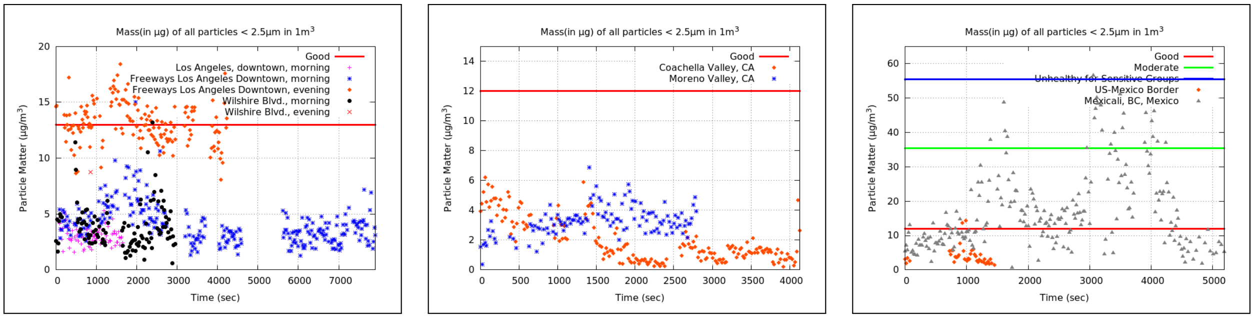
(c) Mexicali, BC

Figure 8.39: Particle Matter of 1 µg/m^3

than in the Inland Empire, which shows a much lower concentration of the greenhouse gas, see Figure 8.37b. We concluded that the urban areas, 8.4.2.1 and 8.4.2.1, have significant higher levels of concentration and are both well above the average global CO_2level measured by NOAA of $400ppm$. Again, the valleys have a much cleaner air than the urban cities.

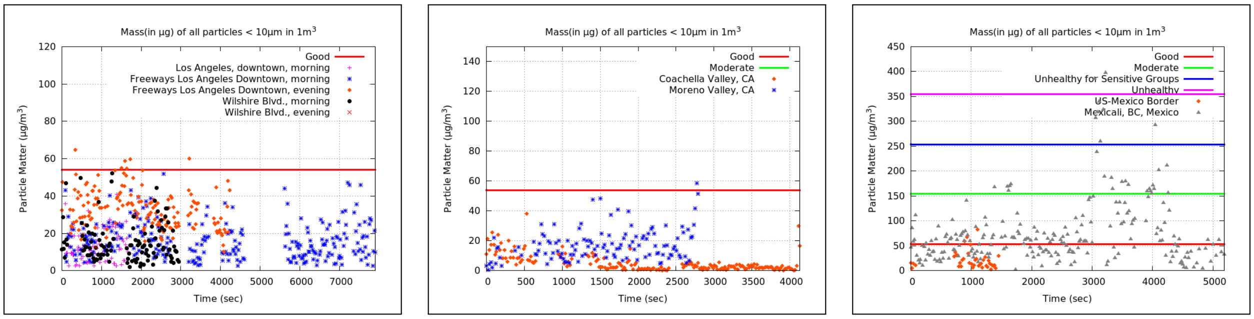
We proceed now to describe our observations of the O_3levels. Figures 8.38a, 8.38b, and 8.36c show similar concentration levels; however, those levels in both Los Angeles and Mexicali tend to dwell in the moderate and unhealthy for sensitive group levels, while the Coachella Valley stays mostly below the moderate. Our conclusion for this greenhouse gas is that the low to medium levels are related mostly to the combustion of vehicle fuel; in other words, its presence is directly related to the presence of vehicles.

Particle matter, especially those with diameter sizes below the $2.5\mu g/m^3$, are very dangerous for the human system because they are very difficult to remove from the respiratory system. Figures 8.39a, 8.39b, and 8.39c show the levels of particles smaller than $1\mu g/m^3$. As the figures



(a) Los Angeles, CA (b) Coachella Valley, CA (c) Mexicali, BC

Figure 8.40: Particle Matter of 2.5 μg/m^3



(a) Los Angeles, CA (b) Coachella Valley, CA (c) Mexicali, BC

Figure 8.41: Particle Matter of 10 μg/m^3

demonstrate, Mexicali shows the largest spread of particles suspended; Los Angeles comes second but with a much more clustered values. Coachella comes third with very low values. Common particles of $1\mu g/m^3$ or less are exhaust gasses from oil and tobacco combustion, soot, and smog; all related to vehicle and factory pollution. Examples of particles of $2.5\mu g/m^3$ include bacteria, house dust, mold spores, cat allergens, etc. Figures 8.40a, 8.40b, and 8.40c show the values of the particles of diameter $2.5\mu g/m^3$ or smaller. On these graphs we can see that while Los Angeles have concentrations above the good category, Mexicali is in fact the most affected of the three; Coachella has values well below the good category. Our conclusions here are that neither of the US cities have a large problem with particle matter below the $2.5\mu g/m^3$ on the road; however, Mexicali does, and that is mostly because the amount of dust contamination on the streets and its proximity to the agriculture fields.

Our last particle, those with diameter $10\mu g/m^3$ or less, are shown in Figures 8.41a, 8.41b, and 8.41c. As with the other particle matters, the concentration amount of particles of diameter

$10\mu\mathrm{g}/\mathrm{m}^3$ or less are mostly below in the US regions; Mexicali is the most affected by the amount of suspended particles. Our conclusion for this is the same as our conclusion for the concentration of the smaller particles: the city is very dusty.

8.5 Conclusion

The measurement of the wireless electromagnetic spectrum, specifically the ISM band, is an important task if we wish to establish a Cognitive Radio Network, or CogNet. A Vehicular Network, or VANET, is a network that operates among the participants of the roads in a urban area. It includes the vehicles, the pedestrians, the electronic equipment of a traffic signals, and any other device connected to the Internet interested in establishing communications with the vehicles. Given the mobile nature of a VANET, it is desirable to combine it with a CogNet in order to adapt to the spectrum interference on the road. Moreover, the vehicles operating on a VANET are the best places to sense the environment pollutants that are present in a city. For these reasons, the objective of this project is two-fold: (1) investigate the interference on the road affecting a VANET, and (2) measure the pollutants present around the vehicles operating on a urban area. To accomplish this, we built two test beds, one that measures the pollution and another that measures the interference. We collected our results on a database and made it available for the public to see via a web site. We concluded that it is possible to establish a wireless communications among vehicles; the interference on the road is not harmful but it is detectable and the network may move around it. Finally, we successfully detect and logged the pollutants and prove the concept of a mobile sensor network operating on top of a VANET as a viable solution to measure contaminants in a large area.

8.6 Relevance to the Thesis

This work provides a study on urban pollution and wireless interference collected by sensors on a vehicle. The data related to wireless interference by WiFi base stations at residential and commercial along the path of a collecting vehicles helps us understand the interference of these base stations on VANETs using WiFi as well. We consider the pollution sensors and communication equipment as a VANET cloud database that can be queried; that is a service that needs to maintain service continuity. As the vehicles move on the city streets, foreign entities may query the vehicles for pollution data. Our previous work on V2I and V2I VANETs may be combined with this pollution sensing platform and leverage MPTCP to handover between WiFi access points in the metropolitan network. The CogNet test bed used in this work is an extension of our heterogeneous networks study at the beginning of this thesis. All combined, this work demonstrates that a service such as pollution measurement, that has a large impact to society, may enjoy continuity with the systems described in our previous work. This is a major goal of this thesis.

CHAPTER 9

Conclusion

The goal of this thesis is to provide mobile users with service continuity and address the path selection problem for MPTCP newly discovered paths. The work on this thesis gradually addresses these issues in incremental steps. First it explores the potential benefits on heterogeneous networks and wireless links in vehicular networks to provide safety protocols at vehicular intersections. Vehicles organize into "platoons" to decrease the message exchange, and using multiple channels, platoon leaders exchange messages among all via WiFi and LTE. Then patent work related to LTE is presented; first, on identifying coverage holes in LTE networks and later on adjusting cellular towers to redistribute network resources to troubled areas. MPTCP is put to test under Vehicle-to-Interface and Vehicle-to-Vehicle scenarios and this thesis concludes that it is a viable protocol that provides service continuity for end-to-end communication despite the highly moving dynamics in VANET scenarios. Offloading is a major advantage over single-path TCP connections and MPTCP leverages on the multiple interfaces to distribute traffic loads when paths become available. The problem with path selection is addressed with CapProbe measurements. This thesis proposes algorithms that detect, measure, and classify paths in MPTCP end-to-end connections where only a single network may be used; when a path is classified as bad, it is not prudent to perform a handover, but if the path is satisfactory, no action is taken and MPTCP continues with the handover. The sensor work in this thesis has larger impact to society; monitoring the environment with vehicles help improve the air quality in urban, congested regions. The wireless interference study also helps understand the impact of residential and commercial WiFi interference in VANET operations. In concluding, MPTCP is a well-suited protocol that facilitates seamless handovers, even in vehicular scenarios, and allows for service continuity; the algorithms developed in this thesis address the path selection problems in MPTCP connections as well.

REFERENCES

[1] Libelium Comunicaciones Distribuidas, S.L. `http://www.libelium.com`.

[2] 3rd. Generation Partnership Project. *3GPP TR 22.803: Feasibility study for Proximity Services (ProSe)*, Relase 12 edition, September 2013.

[3] 3rd. Generation Partnership Project. *3GPP TR 23.890: Optimized offloading to Wireless Local Area Networks (WLAN) in 3GPP Radio Access Technology (RAT) mobility*, Relase 12 edition, September 2013.

[4] 3rd. Generation Partnership Project. Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture description (Release 13). *3GPP TS 36.401 v13.1.0*, March 2016.

[5] S. Al-Sultan, A. H. Al-Bayatti, and H. Zedan. Context-Aware Driver Behavior Detection System in Intelligent Transportation Systems. *IEEE Transactions on Vehicular Technology*, 62(9):4264–4275, November 2013.

[6] Inc. Apple. Use Multipath TCP to create backup connections for iOS, 2016.

[7] B. Arzani, A. Gurney, S. Cheng, R. Guerin, and B. T. Loo. Impact of Path Characteristics and Scheduling Policies on MPTCP Performance. In *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pages 743–748, May 2014.

[8] S. H. Baidya and R. Prakash. Improving the performance of Multipath TCP over Heterogeneous Paths using Slow Path Adaptation. In *2014 IEEE International Conference on Communications (ICC)*, pages 3222–3227, June 2014.

[9] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving tcp/ip performance over wireless networks. In *IN PROCEEDINGS, 1ST ACM CONF. ON MOBILE COMPUTING AND NETWORKING*. ACM, 1995.

[10] S. Balasubramaniam and J. Indulska. Vertical handover supporting pervasive computing in future wireless networks. *Computer Communications*, 27(8):708–719, May 2004.

[11] Jim Barbaresso, Gustave Cordahi, Dominie Garcia, Christopher Hill, Alex Jendzejec, and Karissa Wright. USDOTs Intelligent Transportation Systems (ITS) ITS Strategic Plan 2015-2019. Technical Report FHWA-JPO-14-145, US Department of Transportation, Washington, DC, USA, Decenber 2014.

[12] S. Bitam, A. Mellouk, and S. Zeadally. Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks. *IEEE Wireless Communications*, 22(1):96–102, February 2015.

[13] Uyless Black. *Internet Architecture: An Introduction to IP Protocols*. Prentice Hall, 2000.

[14] Luca Boccassi, Marwan M. Fayed, and Mahesh K. Marina. Binder: A System to Aggregate Multiple Internet Gateways in Community Networks. In *Proceedings of the 2013 ACM MobiCom Workshop on Lowest Cost Denominator Networking for Universal Access*, LCDNet '13, pages 3–8, New York, NY, USA, 2013. ACM.

[15] Jean-Chrysostome Bolot. Characterizing end-to-end packet delay and loss in the internet. *J. High Speed Netw.*, 2(3):305–323, July 1993.

[16] 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 *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, MobiCom '06, pages 50–61, New York, NY, USA, 2006. ACM.

[17] C. Raiciu and M. Handley and D. Wischik. Coupled Congestion Control for Multipath Transport Protocols. RFC 6356, October 2011.

[18] California Department of Motors and Vehicles. Estimated Vehicles Registered by County. `https://www.dmv.ca.gov/portal/wcm/connect/add5eb07-c676-40b4-98b5-8011b059260a/est_fees_pd_by_county.pdf?MOD=AJPERES`, December 2015.

[19] Claudia Campolo, Antonella Molinaro, and Riccardo Scopigno. *Vehicular ad hoc Networks*. Springer, Switzerland, 1 edition, 2015.

[20] Y. C. Chen, Ys. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley. A Measurement-based Study of MultiPath TCP Performance over Wireless Networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 455–468, New York, NY, USA, 2013. ACM.

[21] Y. C. Chen and D. Towsley. On Bufferbloat and Delay Analysis of Multipath TCP in Wireless Networks. In *Networking Conference, 2014 IFIP*, pages 1–9, June 2014.

[22] R. M. N. Chirwa and A. P. Lauf. Performance improvement of transmission in Unmanned Aerial Systems using multipath TCP. In *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 000019–000024, December 2014.

[23] J. Chou and J. Mena. System and method of cell outage compensation in cellular systems, June 2016. US Patent 9363727.

[24] J. Chou and J. Mena. Coverage adjustment in e-utra networks, January 2017. US Patent 9538413, 9154978, 8995255.

[25] J. Chou and J. Mena. Identifying coverage holes using inter-rat handover measurements, May 2017. US Patent 9660710, 8868067.

[26] Claudia Campolo and Antonella Molinaro and Riccardo Sopigno. *Vehicular ad hoc Networks: Standards, Solutions, and Research*. Springer, Switzerland, 2015.

[27] Riccardo Crepaldi, Ryan Beavers, Braden Ehrat, Matt Jaeger, Steven Biersteker, and Robin Kravets. LoadingZones: Leveraging Street Parking to Enable Vehicular Internet Access. In *Proceedings of the Seventh ACM International Workshop on Challenged Networks*, CHANTS '12, pages 23–30, New York, NY, USA, 2012. ACM.

[28] Erik Dahlman, Stefan Parkvall, and Johan Sköld. *4G LTE / LTE-Advanced for Mobile Broadband*. Academic Press, 2011.

[29] Z. Dai, R. Fracchia, J. Gosteau, P. Pellati, and G. Vivier. Vertical Handover Criteria and Algorithm in IEEE802.11 and 802.16 Hybrid Networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 2480–2484, 2008.

[30] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan. WiFi, LTE, or Both?: Measuring Multi-Homed Wireless Internet Performance. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 181–194, New York, NY, USA, 2014. ACM.

[31] Pralhad Deshpande, Xiaoxiao Hou, and Samir R. Das. Performance Comparison of 3G and Metro-scale WiFi for Vehicular Network Access. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 301–307, New York, NY, USA, 2010. ACM.

[32] Kakan Chandra Dey, Anjan Rayamajhi, Mashrur Chowdhury, Parth Bhavsar, and James Martin. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Communication in a Heterogeneous Wireless Network: Performance Evaluation. *Transportation Research Part C: Emerging Technologies*, 68:168 – 184, July 2016.

[33] P. Du, S. Nazari, J. Mena, R. Fan, M. Gerla, and R. Gupta. Multipath TCP in SDN-enabled LEO satellite networks. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 354–359, Nov 2016.

[34] Maeve Duggan. Cell Phone Activities 2013. `http://pewinternet.org/Reports/2013/Cell-Activities.aspx`, September 2013. Web site.

[35] Nandita Dukkipati, Tiziana Refice, Yuchung Cheng, Jerry Chu, Tom Herbert, Amit Agarwal, Arvind Jain, and Natalia Sutin. An Argument for Increasing TCP's Initial Congestion Window. *SIGCOMM Comput. Commun. Rev.*, 40(3):26–33, June 2010.

[36] L. Eastwood, S. Migaldi, Qiaobing Xie, and V. Gupta. Mobility Using IEEE 802.21 in a Heterogeneous IEEE 802.16/802.11-based, IMT-advanced (4G) network. *Wireless Communications, IEEE*, 15(2):26–34, 2008.

[37] Ekram Hossain and Vijay Bhargava. *Cognitive Wireless Communication Networks*. Springer, New York, NY, USA, 2007.

[38] Mustafa Ergen. *Mobile Broadband Including WiMax and LTE*. Springer, 2009.

[39] Federal Communications Commission. Understanding the FCC Regulations for Low-Power, Non-Licensed Transmitters. `https://transition.fcc.gov/Bureaus/Engineering_Technology/Documents/bulletins/oet63/oet63rev.pdf`, February 1994.

[40] Federal Communications Commission. FCC: What We Do. `https://www.fcc.gov/about-fcc/what-we-do`, December 2015.

[41] S. Ferlin, T. Dreibholz, and Ö Alay. Multi-path transport over heterogeneous wireless networks: Does it really pay off? In *2014 IEEE Global Communications Conference*, pages 4807–4813, December 2014.

[42] S. Ferlin-Oliveira, T. Dreibholz, and Ö Alay. Tackling the challenge of bufferbloat in Multi-Path Transport over heterogeneous wireless networks. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, pages 123–128, May 2014.

[43] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural Guidelines for Multipath TCP Development. RFC 6182, March 2011.

[44] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824, January 2013.

[45] M. Forster, R. Frank, M. Gerla, and T. Engel. A Cooperative Advanced Driver Assistance System to Mitigate Vehicular Traffic Shock Waves. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 1968–1976, April 2014.

[46] Matthew S. Gast. *802.11® Wireless Networks*. O'Reilly, 2nd. edition, 2005.

[47] P. Gomes, F. Vieira, and M. Ferreira. The see-through system: From implementation to test-drive. In *2012 IEEE Vehicular Networking Conference (VNC)*, pages 40–47. IEEE, November 2012.

[48] R. Good and N. Ventura. A Multilayered Hybrid Architecture to Support Vertical Handover between IEEE802.11 and UMTS. In *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, IWCMC '06, pages 257–262, New York, NY, USA, 2006. ACM.

[49] Yuci Gou, D. A J Pearce, and P.D. Mitchell. A Receiver-based Vertical Handover Mechanism for TCP Congestion Control. *Wireless Communications, IEEE Transactions on*, 5(10):2824–2833, 2006.

[50] S. Ha, I. Rhee, and L. Xu. CUBIC: A New TCP-friendly High-speed TCP Variant. *SIGOPS Oper. Syst. Rev.*, 42(5):64–74, July 2008.

[51] Georg Hampel, Anil Rana, and Thierry Klein. Seamless TCP Mobility Using Lightweight MPTCP Proxy. In *Proceedings of the 11th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '13, pages 139–146, New York, NY, USA, 2013. ACM.

[52] B. Han, Qian F., Hao S., and L. Ji. An Anatomy of Mobile Web Performance over Multipath TCP. In *2015 11th International Conference on Emerging Networking EXperiments and Technologies*, ACM CoNext 2015, New York, NY, USA, 2015. ACM.

[53] Hannes Hartenstein and Kenneth P. Laberteaux. *VANET: Vehicular Applications and Inter-Networking Technologies*. Wiley, Chichester, West Sussex, UK, 1 edition, 2010.

[54] P. Hurtig and A. Brunstrom. Enhanced metric caching for short TCP flows. In *2012 IEEE International Conference on Communications (ICC)*, pages 1209–1213, June 2012.

[55] IEEE Standards Association. IEEE Standard for Air Interface for Broadband Wireless Access Systems. *IEEE Std 802.16-2012 (Revision of IEEE Std 802.16-2009)*, pages 1–2542, August 2012.

[56] IEEE Standards Association. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 801.11-2012 (Revision of the IEEE Std 802.11-2007)*, pages 1–2793, March 2012.

[57] IEEE Standards Association. IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. *IEEE Std 1609.0-2013*, pages 1–78, March 2014.

[58] Joseph Mitola III. *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*. PhD thesis, Royal Institute of Technology (KTH), 2000.

[59] INEGI: Instituto Nacional de Estadística y Geografía. Censo de Población y Vivienda 2010. `http://www.inegi.org.mx/est/contenidos/proyectos/ccpv/`, December 2015.

[60] International Telecomunication Union. Article 1: Terms and Definitions. `http://life.itu.int/radioclub/rr/art01.htm`, October 2009.

[61] Université Catholique de Louvain Internet Network Lab. MultiPath TCP - Linux Kernel Implementaiton. `http://multipath-tcp.org`. Web site.

[62] ITU. International Telecommunication Union. `http://www.itu.int/`. Web site.

[63] ITU. International Mobile Telecommunications-2000 (IMT-2000). `http://www.itu.int/rec/R-REC-M.687-2-199702-I/en`, February 1997. Web page.

[64] ITU. Report ITU-R SM.2152: Definitions of Software Defined Radios (SDR) and Cognitive Radio System (CRS). Technical report, International Telecommunication Union, September 2009.

[65] ITU. Detailed specifications of the terrestrial radio interfaces of International Mobile Telecommunications Advanced (IMT-Advanced). `http://www.itu.int/rec/R-REC-M.2012-0-201201-I/en`, January 2012. Web page.

[66] Konstantin Ivanov. Linux TCP Tuning, September 2010.

[67] M. Jerbi, P. Marlier, and S. M. Senouci. Experimental Assessment of V2V and I2V Communications. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6, October 2007.

[68] Jiang, Haiqing and Wang, Yaogong and Lee, Kyunghan and Rhee, Injong. Tackling Bufferbloat in 3G/4G Networks. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 329–342, New York, NY, USA, 2012. ACM.

[69] editor Jon Postel. Transmission Control Protocol Specification. RFC 793, September 1981.

[70] JSON. Introducing JSON. `https://www.json.org`.

[71] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M. Y. Sanadidi. Capprobe: A simple and accurate capacity estimation technique. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 67–78, New York, NY, USA, 2004. ACM.

[72] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M. Y. Sanadidi. Capprobe: A simple and accurate capacity estimation technique. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 67–78, New York, NY, USA, 2004. ACM.

[73] Rohit Kapoor, Ling-Jyh Chen, Li Lao, Mario Gerla, and M. Y. Sanadidi. Capprobe: A simple and accurate capacity estimation technique. SIGCOMM'04, pages 67–78, New York, NY, USA, 2004. ACM.

[74] Meriem Kassar, Brigitte Kervella, and Guy Pujolle. An overview of vertical handover decision strategies in heterogeneous wireless networks. *Computer Communications*, 31(10):2607 – 2620, 2008.

[75] J. B. Kenney. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182, July 2011.

[76] Srinivasan Keshav. A control-theoretic approach to flow control. *SIGCOMM Comput. Commun. Rev.*, 21(4):3–15, August 1991.

[77] R. Khalili, N. Gast, M. Popovic, U. Upadhyay, and J. Y. Le Boudec. MPTCP is Not Pareto-optimal: Performance Issues and a Possible Solution. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, pages 1–12, New York, NY, USA, 2012. ACM.

[78] Farooq Khan. *LTE for 4G Mobile Broadband.* Cambridge University Press, 2009.

[79] Won-Ik Kim, Bong-Ju Lee, Jae-Su Song, Yeon-Seung Shin, and Yeong-Jin Kim. Ping-Pong Avoidance Algorithm for Vertical Handover in Wireless Overlay Networks. In *VTC Fall07*, pages 1509–1512, 2007.

[80] James F. Kurose and Keith W. Ross. *Computer Networking: A Top-Down Approach.* Addison-Wesley, 5th. edition, 2010.

[81] M. S. Lee, W. T. Kim, J. B. Lee, and Y. W. Lee. Architectural perspective on collaborative multipath TCP in mobile environment. In *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*, pages 475–478, August 2015.

[82] William C. Y. Lee. *Wireless & Cellular Telecommunications*. McGraw Hill, 3rd. edition, 2005.

[83] Adam Lella. comscore reports september 2013 u.s. smartphone subscriber market share. `http://www.comscore.com/Insights/Press_Releases/2013/11/comScore_Reports_September_2013_U.S._Smartphone_Subscriber_Market_Share`, November 2013. Web article.

[84] M. Li, A. Lukyanenko, S. Tarkoma, Cui Y., and A. Ylä-Jääski. Tolerating Path Heterogeneity in Multipath TCP with Bounded Receve Buffers. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '13, pages 375–376, New York, NY, USA, 2013. ACM.

[85] Lien-Wu Chen and Yu-Chee Tseng and Kun-Ze Syue. Surveillance on-the-road: Vehicular tracking and reporting by V2V communications. *Computer Networks*, 67:154 – 163, 2014.

[86] Li Ma, F. Yu, V. C M Leung, and T. Randhawa. A New Method to Support UMTS/WLAN Vertical Handover Using SCTP. *Wireless Communications, IEEE*, 11(4):44–51, 2004.

[87] M.Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581, April 1999.

[88] Johann Márquez-Barja, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. An overview of vertical handover techniques: Algorithms, protocols and tools. *Computer Communications*, 34(8):985 – 997, 2011.

[89] Marshall Sittig. *Particulates and Fine Dust Removal*. Noyes Publications, 1977.

[90] Marshall Sittig. *Handbook of Toxic and Hazardous Chemicals and Carcinogens*. Noyes Publications, 1985.

[91] Matthew Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly, 2005.

[92] Jorge Mena, Peter Bankole, and Mario Gerla. Multipath TCP on a VANET: A Performance Study. *SIGMETRICS Perform. Eval. Rev.*, 44(1):39–40, June 2017.

[93] Jorge Mena, Mario Gerla, and Vana Kalogeraki. Mitigate Funnel Effect in Sensor Networks with Multi-interface Relay Nodes. In *Distributed Computing in Sensor Systems (DCOSS), 2012 IEEE 8th International Conference on*, pages 216–223, 2012.

[94] Jorge Mena, Mario Gerla, and Félix Fernando González Navarro. Vehicular Network Operating on a Cognitive Network. Technical report, University of California Institute for Mexico and the United States, UCMEXUS, Riverside, CA, USA, March 2016.

[95] Jorge Mena and Vana Kalogeraki. Dynamic Relay Node Placement in Wireless Sensor Networks. In *Applications and the Internet, 2008. SAINT 2008. International Symposium on*, pages 8–17, 2008.

[96] Multipath-TCP. Linux Kernel Implementation; v0.90, June 2015.

[97] Ben Munson. Comcast, TWC CableWiFi shared network deal survives recent M&A, hotspot count now 500K, July 2016.

[98] S. C. Nguyen, X. Zhang, T. M. T. Nguyen, and G. Pujolle. Evaluation of Throughput Optimization and Load Sharing of Multipath TCP in Heterogeneous Networks. In *2011 Eighth International Conference on Wireless and Optical Communications Networks*, pages 1–5, May 2011.

[99] Sinh Chung Nguyen, Xiaofei Zhang, Thi-Mai-Trang Nguyen, and G. Pujolle. Evaluation of throughput optimization and load sharing of multipath tcp in heterogeneous networks. In *Wireless and Optical Communications Networks (WOCN), 2011 Eighth International Conference on*, pages 1–5, 2011.

[100] Institute of Electrical and Electronics Engineers. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pages 1–2793, 2012.

[101] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. Exploring Mobile/WiFi Handover with Multipath TCP. In *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, CellNet '12, pages 31–36, New York, NY, USA, 2012. ACM.

[102] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure. Experimental Evaluation of Multipath TCP Schedulers. In *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*, CSWS '14, pages 27–32, New York, NY, USA, 2014. ACM.

[103] Paper, Cisco White. Cisco Visual Networking Index: Forecast and Methodology, 2015-2020. February 2016.

[104] C. Pollalis, P. Charalampou, and E. Sykas. HTTP Data Offloading Using Multipath TCP Proxy. IEEE CIT15, pages 777–782, October 2015.

[105] Editor R. Stewart. Stream Control Transmission Protocol. RFC 4960, September 2007.

[106] C. Raiciu, Paasch C., Barre S., Ford A., Honda M., Duchene F., Bonaventure O., and M. Handley. How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 399–412, San Jose, CA, 2012. USENIX.

[107] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley. Opportunistic Mobility with Multipath TCP. In *Proceedings of the Sixth International Workshop on MobiArch*, MobiArch '11, pages 7–12, New York, NY, USA, 2011. ACM.

[108] Karunaharan Ratnam and Ibrahim Matta. Wtcp: An efficient mechanism for improving wireless access to tcp services. *International Journal of Communication Systems*, 16:47–62, 2003.

[109] H. Rutagemwa, S. Pack, Xuemin Shen, and Jon W. Mark. Robust Cross-Layer Design of Wireless-Profiled TCP Mobile Receiver for Vertical Handover. *Vehicular Technology, IEEE Transactions on*, 56(6):3899–3911, 2007.

[110] M. Scharf and A. Ford. Multipath TCP (MPTCP) Application Interface Considerations. RFC 6897, March 2013.

[111] SEMARNAT: Secretaría del Medio Ambiente y Recursos Naturales. Programa Para Mejorar La Calidad del Aire en Mexicali 2011-2020. http://www.semarnat.gob.mx/temas/gestion-ambiental/calidad-del-aire/programas-de-gestion-para-mejorar-la-calidad-del-aire, 2014.

[112] Stefania Sesia, Issam Toufik, and Matthew Baker. *LTE – The UMTS Long Term Evolution*. Wiley, 2009.

[113] Society of Automotive Engineers. Dedicated Short Range Communications (DSRC) Message Set Dictionary. *SAE Std. J2735*, March 2016.

[114] Travis Sparks. How To: Network/TCP/UDP Tuning, 2003.

[115] Simon E. Spero. Analysis of HTTP Peroformance problems, July 1994.

[116] Stephan Olariu and Michele C. Weigle. *Vehicular Networks: From Theory to Practice*. CRC Press, Boca Raton, FL, USA, 2009.

[117] L. C. Tung, J. Mena, M. Gerla, and C. Sommer. A cluster based architecture for intersection collision avoidance using heterogeneous networks. In *2013 12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, pages 82–88, June 2013.

[118] US Census Bureau. 1 Million Milestone. https://www.census.gov/content/dam/Census/newsroom/releases/2015/cb15-89_graphic.jpg, July 2014.

[119] US Enviromental Protection Agency. Criteria Air Pollutants. https://www.epa.gov/criteria-air-pollutants.

[120] US Environmental Protection Agency. The Plain English Guide To The Clean Air Act. https://www3.epa.gov/airquality/peg_caa/, April 2007.

[121] US Senate. The Clean Air Act (42 U. S. C. 7401-7626). http://www.epw.senate.gov/envlaws/cleanair.pdf, February 2004.

[122] Y. Wang, J. Jiang, and T. Mu. Context-Aware and Energy-Driven Route Optimization for Fully Electric Vehicles via Crowdsourcing. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1331–1345, September 2013.

[123] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, Implementation and Evaluation of Congestion Control for Multipath TCP. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 99–112, Berkeley, CA, USA, 2011. USENIX Association.

[124] Damon Wischik, Mark Handley, and Marcelo Bagnulo Braun. The Resource Pooling Principle. *SIGCOMM Comput. Commun. Rev.*, 38(5):47–52, September 2008.

[125] Chwan-Hwa (John) Wu and J. David Irwin. *Introduction to Computer Networks and Cybersecurity*. CRC Press, 2013.

[126] C. Xu, F. Zhao, J. Guan, H. Zhang, and G. M. Muntean. Qoe-driven user-centric vod services in urban multihomed p2p-based vehicular networks. *IEEE Transactions on Vehicular Technology*, 62(5):2273–2289, June 2013.

[127] L. Xu, K. Harfoush, and I. Rhee. Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2514–2524 vol.4, March 2004.

[128] Xiaohuan Yan, Y. Ahmet ekerciolu, and Sathya Narayanan. A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks. *Computer Networks*, 54(11):1848 – 1863, 2010.

[129] F. Yang, P. Amer, and N. Ekiz. A Scheduler for Multipath TCP. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, July 2013.

[130] R. Yu, Y. Zhang, S. Gjessing, W. Xia, and K. Yang. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, 27(5):48–55, September 2013.

[131] D. Zhou, W. Song, and P. Ju. Subset-sum Based Relay Selection for Multipath TCP in Cooperative LTE networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 4735–4740, December 2013.

[132] Dizhi Zhou, Wei Song, and Minghui Shi. Goodput improvement for multipath tcp by congestion window adaptation in multi-radio devices. In *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, pages 508–514, 2013.