Experimental Analysis of Opportunistic Communication for Vehicular Internet Access

by

David Hadaller

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis examines the problem of using 802.11 hotspots for vehicular Internet access. In this access paradigm, a user in a vehicle performs batch transfers by opportunistically communicating with roadside 802.11 access points while driving along a highway. Despite the short connection duration, a significant amount of data can be transferred. Because complete coverage is not needed, this method of Internet access provides a low-cost alternative to using cellular technology for applications that can tolerate some delay and require large data transfer such as sending or receiving music, movies, or digital photographs.

Although vehicular opportunistic connections offer the potential to transfer a large of amount of data, utilizing this potential is non-trivial because existing transport and data-link layer network protocols were not designed for this use.

This thesis presents an experimental analysis of transport and data-link layer protocol operation at a level of detail not previously explored. We identify ten problems that cause a reduction of up to 50% of the amount of data that could have been transferred in this scenario. Our primary finding is that transmission errors during connection setup and inadequate MAC data rate selection are the main causes of the underutilization of the connection. Based on these findings we make preliminary recommendations for best practices for using vehicular opportunistic connections. In particular, we argue that overall throughput could be significantly improved if environmental information was available to the lower layer network protocols.

Acknowledgements

This work first began as a course project in a course offered by Professor Keshav, who initially suggested a project to communicate data with a race car as it rounded the track. My group members in this project, Herman Li and L.G. Alex Sung, deserve thanks for getting the ball rolling on what ended up becoming my thesis.

For adequately distracting me in the lab, I thank my friends and lab-mates Nabeel Ahmed, Aaditeshwar Seth, Shimin Guo, Earl Oliver, Hossein Falaki, and Sumair Rahman. Nabeel, in addition to being a distraction, provided much necessary feedback at any hour of the day.

The student interns that I had the opportunity to work with, Shubham Agarwal and Aditya Nayar, deserve thanks for providing assistance with work that became part of this thesis.

Without my supervisors, Tim Brecht and Srinivasan Keshav, none of this work would have developed as far as it did. The guidance, challenge, and drive they provided was invaluable throughout my graduate program.

Thanks goes to my thesis readers, Martin Karsten and Paul Ward, as well as the other faculty and graduate students in the Networks and Distributed Systems research group who helped refine both my research material and my presentation skills.

I would like to thank Kimberly Tsang for her assistance running experiments and her unending care and support throughout my degree.

Lastly, I gratefully acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC), the David R. Cheriton School of Computer Science, and the University of Waterloo for the financial support that made this work possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we consider the problem of obtaining Internet access in vehicles using short-lived connections to roadside 802.11 access points that arise opportunistically as vehicles are in motion. To better understand this new paradigm of access, we perform a detailed experimental analysis of the behavior of data-link and transport layer protocols during opportunistic communication at a depth not previously explored.

This connectivity paradigm can be used for (1) downloading data to the vehicle, such as for passengers to preview movie trailers as they travel to the cinema, to download product-specific promotional information from local retailers as a family drives between shopping outlets searching for a particular item, or to provide traffic and weather updates for navigational purposes; (2) uploading data from the vehicle, such as for unloading digital camera images or video on a road trip; and (3) transshipment of data, including relaying sensor data from disconnected stations to infrastructure nodes, as done by Seth et al. [59].

Connecting wirelessly to the Internet while moving can be done using two common technologies: through a cellular network or through an 802.11 hotspot, which is an 802.11 Wi-Fi access point with a backhaul connection to the Internet. Cellular networks provide near-ubiquitous coverage in most cities in the world, while 802.11 hotspots are confined to smaller coverage areas and are typically located in homes or businesses such as coffee shops, gas stations, and libraries. Because a hotspot's backhaul connection could be a bottleneck, we assume the access point has some storage that allows it to act as a cache between wireless users and the Internet.

Using a cellular network to transfer data comes at a much higher cost per bit than using an 802.11 hotspot. Due to the need for complete coverage, cellular systems are both more expensive and offer lower data rates than 802.11 hotspots.

Cost: Cellular systems are designed primarily for voice and therefore aim for anytime, anywhere service. To setup and maintain the significant infrastructure required for such near-ubiquitous coverage comes at great cost, often exceeding hundreds of millions of dollars [17]. A single 802.11 hotspot, on the other hand, requires only a commodity access point and an inexpensive cable or DSL backhaul connection. The number of hotspots required for satisfactory coverage depends on user's needs and type of application.

Data Rate: Because the signal does not need to travel as far, 802.11 has been designed to transmit data using a more dense encoding scheme, and thus a higher data rate, than cellular technology. 802.11a [31], released in 1999, and 802.11g, released in 2003, support peak link layer rates of up to 54 Mbps and typical goodput rates of approximately 30 Mbps, as shown later in Chapter 4. CDMA cellular technology using 1X EVDO, standardized in 1999, supports peak link layer rates of up to 2.4 Mbps downlink and 153 Kbps uplink, but TCP goodput in practical scenarios has been measured to be only 573 Kbps downlink and 95 Kbps uplink [40]. GPRS EDGE cellular networks, first available in 2003, support peak link layer rates of up to 474 Kbps downlink but TCP goodput of only 32 Kbps has been measured in practical scenarios [11]. Newer technologies, such as 802.11n, 802.16 WiMAX, and 4G cellular, all support higher data rates; however, the tradeoff between coverage and data rate remains.

Therefore, 802.11 hotspots provide much lower cost per bit than cellular systems, at the expense of reduced coverage. Many common applications, such as email, transferring pictures or video, and bulk downloads, can already tolerate significant outages between periods of connectivity. This relieves the need for ubiquitous coverage, allowing for a much less expensive access paradigm that still provides the ability to transfer large amounts of data. In our experiments, in Chapter 4, a vehicle traveling past an 802.11 roadside access point at 80 km/h is able to transfer up to 50 MB of data, the equivalent of approximately 20 songs, 15 minutes of low quality video, or 125 high quality digital photographs.

Despite this potential, making effective use of opportunistic connections to 802.11 access points while moving is non-trivial due to the short connection duration and rapid fluctuations in signal quality. In this thesis, we examine the problem of effectively using opportunistic connections by analyzing the shortcomings of existing data-link and transport layer protocols and making preliminary recommendations to best use opportunistic connections, as discussed next.

1.1 Utilizing Opportunistic Connections

Previous work [49, 27, 10, 18, 42] has confirmed the feasibility of using opportunistic communication for vehicular Internet access. In this thesis, we build on previous work with a detailed experimental analysis at a depth not previously explored. Our empirical analysis of opportunistic vehicular connections reveals the following three problem areas:

- As a vehicle enters the range of the access point, wireless losses at the fringe of access point coverage are not handled well during the connection setup phase, causing both the 802.11 MAC and TCP protocols to enter a back-off state that can last into the useful period of the connection.

- Once the useful period of the connection has begun, the data rate selection algorithm at the MAC layer is not dynamic enough to achieve the best possible throughput in such a rapidly changing environment.
- When leaving the coverage area, the 802.11 MAC and TCP protocols can again respond to wireless losses at the fringe by performing a back-off procedure, resulting in a further decrease in throughput.

We argue that lack of environmental awareness is the fundamental underlying cause of these problems. We observed that signal quality was highly correlated with position on the road. Had TCP and the 802.11 MAC layer known that they were in an environment where the signal strength first increases and then decreases, with reasonably predictability, they could have adapted their behavior to better suit the environment.

Our work describes a scenario where heightened awareness of the environment can improve overall throughput. We argue that this insight can be broadly applied to communication protocols in general. That is, environmental information can be used to: (1) choose more suitable initial operating parameters, such as the initial TCP timeout value or the initial MAC data rate and (2) tune protocol behavior to better handle the amount of packet loss or delay at any point in time.

The majority of our analysis of vehicular opportunistic communication in this thesis was presented previously at the ACM MobiSys 2007 conference [26] and the details of our GPS analysis, in Section 3.5 of this thesis, were presented in the MobiEval workshop [24] as part of the same conference.

1.2 Contributions

Our contributions can be summarized as follows:

1. **We experimentally demonstrate that data-link and transport layer protocols achieve only half of the available potential throughput**. During a single pass of a roadside access point at 80 km/h, we found that protocol behavior significantly inhibited the amount of data that was transferred.
2. **We experimentally identify a complex interplay of ten distinct causes of reduced data transferred and quantify the impact of each**. In particular, even with a streamlined connection setup procedure that does not use DHCP, delayed connection setup due to (a) lengthy access point selection, (b) MAC management timeouts, (c) ARP timeouts, (d) poor MAC bit rate selection, and (e) TCP timeouts, results in a loss of nearly 25% of

overall throughput 15% of the time, and up to a loss of 40% in the worst case. In addition to problems at the start of the connection, we show that further problems during the connection and near its end also contribute to lost throughput.

3. **We make preliminary recommendations for best practices for using vehicular opportunistic connections**. Based on our experimental findings of how current data-link and transport layer protocols underutilize connection potential, we suggest ways in which heightened awareness of the operating environment could be used to increase the overall throughput of a vehicular connection.

1.3 Thesis Organization

This thesis is organized as follows. After discussing related work in Chapter 2, we describe our experimental setup in Chapter 3, including details of how we address systematic and random error in GPS measurements in Section 3.5. Following that we detail our experimental results in Chapter 4, beginning with some key initial observations. We then show the extent to which current data-link and transport layer protocols underutilize vehicular connections in Section 4.2, followed by a detailed look at root causes in Section 4.3. Based on our experimental findings we then make preliminary recommendations for best practices for vehicular opportunistic connections in Section 4.4 and discuss how our work shows the broader benefits of environmental awareness for communication protocols in general in Section 4.5. Finally, we present conclusions and future work in Chapter 5.

Chapter 2

Background and Related Work

Using intermittently connected links for communication has been studied in many different contexts. Figure 2.1 gives an overview of the related work and can be used as a road-map for this chapter.

Using opportunistic connections as a means of communication presents many different challenges at several levels. In general, networking protocols have been designed with the assumption of a single persistent connection and have a limited ability to handle intermittent connectivity and changing points of attachment.

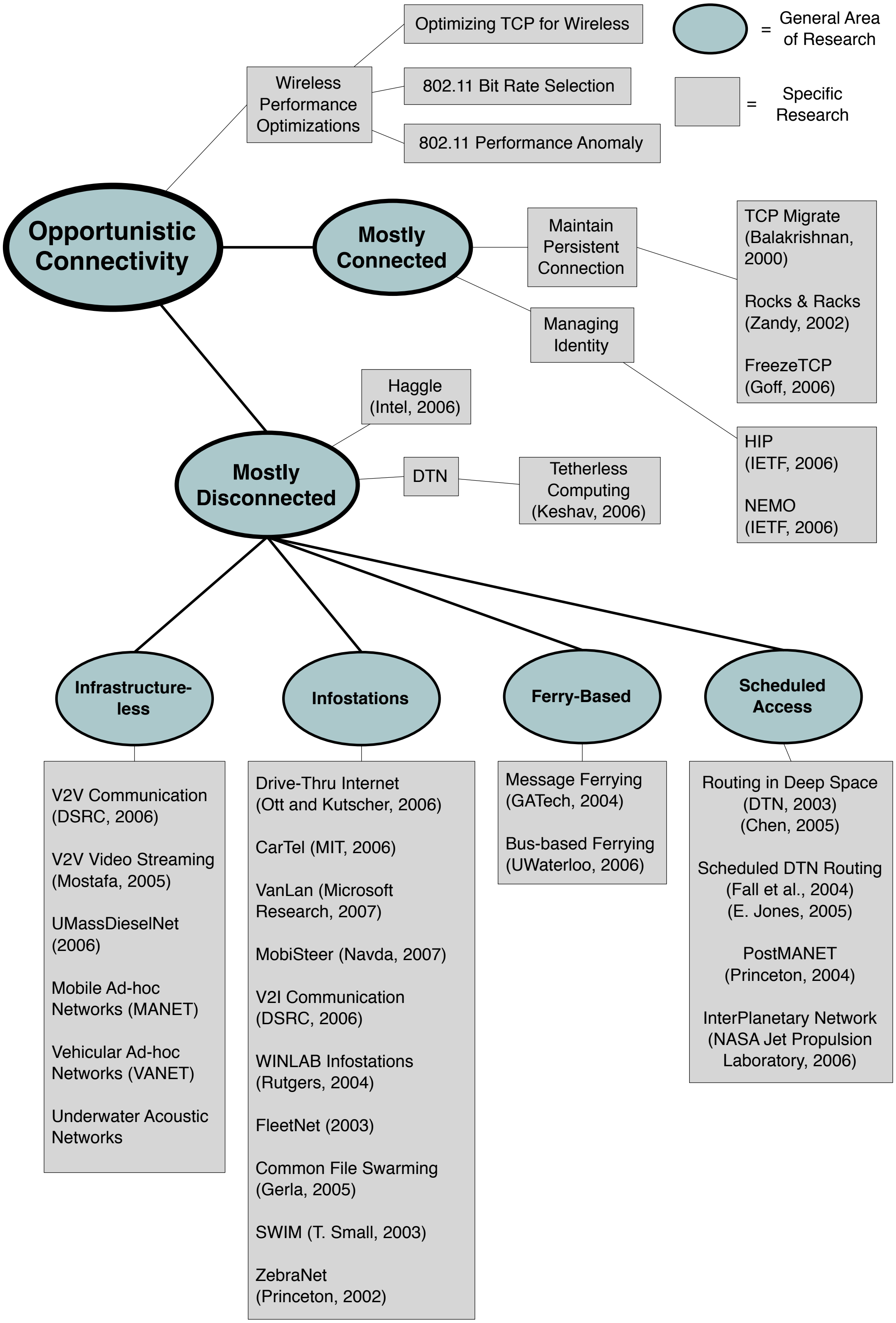
From an architectural point of view, managing a user's identity across multiple connections, routing data between intermittent users, and ensuring reliable end-to-end delivery are all challenges that must be addressed when using opportunistic connectivity. For each individual connection, it is important to setup a connection quickly and make best use of the connection time while fairly sharing access to the wireless medium.

Opportunistic communication can be broken down into two broad categories, as shown in Figure 2.1: *Mostly Connected* (Sec. 2.1), which assumes connectivity will be available almost all of the time with only short and infrequent outages, and *Mostly Disconnected* (Sec. 2.2), which assumes connectivity will only occasionally be available. The key difference is that in the former case, networking protocols need only to mask the temporary disconnections from the user, as opposed to the more challenging case involving longer periods of disconnection.

The work in this thesis falls primarily into the category of mostly disconnected; however, any scenario involving the use of opportunistic connections may benefit from the insights gained from the experimental analysis performed in this thesis.

2.1 Mostly Connected

If connectivity can be assumed to be available most of the time, then transparent operation can be provided by simply masking the brief periods of disconnection from



= General Area of Research
= Specific Research
Opportunistic Connectivity
Wireless Performance Optimizations
Optimizing TCP for Wireless
802.11 Bit Rate Selection
802.11 Performance Anomaly
Mostly Connected
Maintain Persistent Connection
Managing Identity
TCP Migrate (Balakrishnan, 2000)
Rocks & Racks (Zandy, 2002)
FreezeTCP (Goff, 2006)
HIP (IETF, 2006)
NEMO (IETF, 2006)
Mostly Disconnected
Haggle (Intel, 2006)
DTN
Tetherless Computing (Keshav, 2006)
Infrastructure-less
Infostations
Ferry-Based
Scheduled Access
V2V Communication (DSRC, 2006)
V2V Video Streaming (Mostafa, 2005)
UMassDieselNet (2006)
Mobile Ad-hoc Networks (MANET)
Vehicular Ad-hoc Networks (VANET)
Underwater Acoustic Networks
Drive-Thru Internet (Ott and Kutscher, 2006)
CarTel (MIT, 2006)
VanLan (Microsoft Research, 2007)
MobiSteer (Navda, 2007)
V2I Communication (DSRC, 2006)
WINLAB Infostations (Rutgers, 2004)
FleetNet (2003)
Common File Swarming (Gerla, 2005)
SWIM (T. Small, 2003)
ZebraNet (Princeton, 2002)
Message Ferrying (GATech, 2004)
Bus-based Ferrying (UWaterloo, 2006)
Routing in Deep Space (DTN, 2003) (Chen, 2005)
Scheduled DTN Routing (Fall et al., 2004) (E. Jones, 2005)
PostMANET (Princeton, 2004)
InterPlanetary Network (NASA Jet Propulsion Laboratory, 2006)

Figure 2.1: Classification of Related Work

the user. The proposed approach to this is to fool the transport or application layer protocols into thinking that an underlying link layer connection is always present, although it may be constantly changing. The user experience would be much like a persistent connection but with short spikes in response time during the brief periods of disconnection. Work in this area has focused on two main problems.

Maintaining a persistent TCP connection: One of TCP's well-known problems is that it interprets any loss as an indication of network congestion and accordingly backs off [4], as depicted in Figure 2.2. Thus, if a user is briefly disconnected it is important to ensure that TCP does not unnecessarily enter into a deep backoff state. Freeze-TCP [20] was proposed to pause TCP's state if an impending disconnection or handoff were detected by the mobile device, resuming TCP's state once connectivity is reestablished. TCP Migrate [62] and Rocks and Racks [74] have similar goals to Freeze-TCP but require both client and server modification, whereas Freeze-TCP only relies on modification of the mobile's TCP behavior. In mostly disconnected scenarios, attempting to maintain a TCP connection between periods of connectivity is often undesirable as disconnected episodes could last from minutes to days.

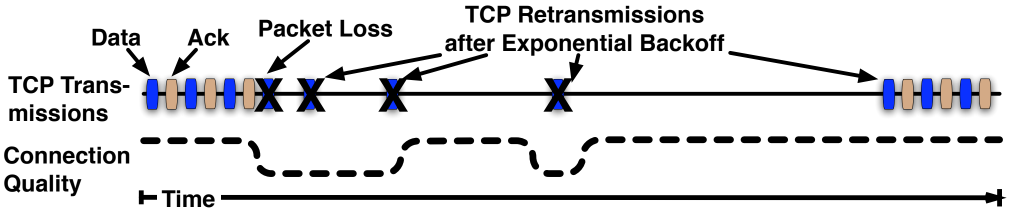


Figure 2.2: The harmful effects of TCP misinterpreting wireless loss as network congestion [4].

Managing identity across points of attachment: As points of attachment change, end-host IP addresses change. Although it is possible to use a middle-box to translate between old and new addresses, such as using MobileIP [32], the IETF is working toward a long-term solution, which would be to associate globally unique identifiers with hosts rather than just IP addresses that are coupled with the network attachment point. The Host Identity Protocol (HIP) [29] by the IETF uses public keys and DNS to avoid the use of a middle-box. Another group in the IETF is examining a similar identity problem involving managing the identity of an entire network that moves and changes attachment points, such that individual users are continually reachable and mobility is transparent to the users. Their approach uses a middle-box for translation and is called NEtwork MObility (NEMO) [48]. Managing identity is also important in mostly disconnected scenarios. Seth et al. [59], in their Tetherless Computing Architecture, propose a scheme that uses globally unique IDs for use in such scenarios.

2.2 Mostly Disconnected

We divide related work on mostly disconnected scenarios into four areas, as shown in the roadmap Figure 2.1: (1) **Infrastructure-less** (Sec. 2.2.1) communication between mobile nodes without the help of infrastructure or a ferrying mechanism; (2) Communication with **Infostations** (Sec. 2.2.2), which are gateways between mobile devices and a backhaul network such as the Internet; (3) **Ferry-Based** (Sec. 2.2.3) where specific message carriers assist with the delivery of messages between nodes and infrastructure; and (4) **Scheduled Access** (Sec. 2.2.4), which is a special case of the first three, having the unique property that the schedules of future links are precisely known, or it is assumed that node connectivity follows a regular pattern that can be learned.

Much of this research falls under the umbrella of **Delay Tolerant Networking (DTN)** [15], which has the general goal of accommodating all types of disconnected operations. In general, DTN operates using the store-and-forward principle, where one mobile node may carry a **bundle** of data from another node in the hope that the data will eventually reach its destination. However, the focus of most of this research has been on routing protocols, rather than on maximizing data transfer during a single opportunistic connection. Our work complements this research.

The Tetherless Computing Architecture [59], built on top of DTN, can be viewed as an extension to DTN and has the goal of providing mobility and disconnection transparency to the mobile user, while using low control overhead and being Internet-compatible and secure.

Furthermore, a separate solution with similar goals is the Haggle project [58]. Haggle aims to eliminate relying on infrastructure unnecessarily when performing tasks such as sending an email between two users who could communicate directly. Haggle's approach to this problem is to "raise" the API to allow applications to specify more meaningful information to the network such as information about the endpoint's identifier, security, and delivery targets or deadlines. Haggle, like DTN, has focused on architectural issues rather than specific network protocol interaction during opportunistic communication as is done in this thesis.

The work in this thesis falls primarily in the infostations area of research; however, there is significant overlap between work done in all areas of mostly disconnected scenarios. Our work focuses on the last hop of the Internet connection, the wireless connection between the vehicle and the access point. We assume that the access point has local storage and can act as a cache between the vehicle and the Internet.

2.2.1 Infrastructure-less

Networks in which communication is between a collection of mobile nodes that are not all connected to one another is referred to as Mobile Ad-hoc Networks

(MANETs) and in the specific case where the mobile nodes are vehicles, they are referred to as Vehicular Ad-hoc Networks (VANETs). MANETs are the subject of much study, the details of which are outside the scope of this thesis.

VANETs: VANETs have received significant research attention due to the unique challenges associated with traveling at high speeds. Guo et al. [23] examined the problem of streaming video from one vehicle to another, using intermediate vehicles to relay the video from source to destination. Another project in this area, UMassDieselNet [66], involved placing small wireless devices on 40+ buses and measuring the inter-contact time and amount of data that could be transferred between buses. Using this data they were able to determine which DTN routing protocols would be most effective in this scenario. They further examined the effectiveness of adding roadside "throw boxes" to improve the number of messages delivered between buses [6]. A throw box is a stationary wireless device that performs a function similar to an intermediary postal mail box. Data can be dropped off at the throw box by one bus for future pick by another bus. The work in this thesis complements this research as network protocol operation during opportunistic communication between buses is very similar to that between a vehicle and a roadside access point.

Dedicated Short Range Communication (DSRC): The DSRC group [14] aims to use short range communication to improve the safety of vehicles on the road. Suppose a driver is making a right turn onto a highway at a stop sign and his/her view is obstructed. With DSRC technology, a warning would be sent from oncoming vehicles announcing their presence to warn the driver not to pull out in front of them. The DSRC group is a US government initiative, backed by many major automobile manufacturers, that will be incrementally rolled out between 2010 and 2020. The communications standard, being developed by the IEEE 802.11p working group, uses 75 MHz of spectrum in the upper 5 GHz range, allocated by the FCC for the purpose of improving traffic safety. The project is based on 802.11a technology and allows some room for commercial applications such as automated payment at a restaurant drive-thru. Although most DSRC work is internal, some work has been published, addressing such problems as location-based broadcast [72] and vehicle-to-vehicle multi-hop communication [43].

A similar initiative is taking place in Japan, termed InternetCAR [65]. Both InternetCAR and DSRC research has focused primarily on safety issues requiring fast response time rather than our focus on bulk transfers in this thesis.

2.2.2 Infostations

Opportunistic communication with roadside gateways was first termed "infofueling" by Frenkiel et al. [17], where a mobile device "fills up" with information at an "infostation". This is similar to how a vehicle gets gas at a gas station.

Early Infostations Work: The idea of using many-time many-where connectivity was first proposed by the WINLAB research center at Rutgers [17]. Their research focuses primarily on motivating the idea and integration with cellular networks [9]. Work at the MAC layer includes improving packet delivery probability through the use of an adaptive retransmission scheme [30]. This work evolved into the technology transfer phase and has become an emergency response project, the details of which are unpublished. Another early project was FleetNet [16] which aimed both to provide ad-hoc communication between vehicles as well as Internet access through multi-hop communication between vehicles and a roadside Internet gateway. Funding for the FleetNet project has since ended and the project no longer exists.

Drive-thru Internet: The first work in infostation-style communication at a practical and experimental level was done by Ott and Kutscher [49] who coined the term Drive-Thru Internet. Their goal was to achieve Internet access in vehicles through opportunistic communication with roadside 802.11 access points. They found that using off-the-shelf 802.11b hardware, a vehicle could maintain a connection to a roadside access point for 600 m, and transfer 9 MB of data at 80 km/h using either TCP or UDP. They found that connections pass through three phases: the entry phase, the production phase, and the exit phase, each lasting 200 m in their experiments. In more recent experiments with 802.11g [50], they were able to transfer between 30 and 70 MB of data at 100 km/h using external antennas.

In order to maintain a persistent connection between a vehicle and an end-host on the Internet, Ott and Kutscher propose the Persistent Connection Management Protocol (PCMP) [50] which uses a proxy to mask the vehicle's mobility to end-hosts as well as to allow vehicles to resume connections established during previous connection opportunities. Further work on Drive-thru Internet has been done on service maps [54], mobile access gateway [51], automating hotspot authentication [52], and bundling the web for DTN [53].

Ott and Kutscher do not examine the specific consequences of data-link and transport layer protocol behavior in their work, as done in this thesis. In their feasibility study they conclude that connection setup must complete before the production phase begins in order to fully utilize the connection [49]. However, they suggest that existing networking protocols are not likely to perform well in the presence of high packet loss and that further investigation is needed to determine the actual impact of the communication characteristics of the entry phase on the overall connection throughput. This is precisely the analysis we perform in this thesis.

Infostations Measurements: Work by Gass et al. [18], termed in-motion networking, has further confirmed the feasibility of using opportunistic connections to vehicles under a variety of different conditions. They study TCP bulk traffic, UDP bulk traffic, and web traffic using standard laptops with no external antennas at speeds ranging from 5 km/h to 120 km/h with various induced backhaul bandwidth and delay parameters. In general, they found that the more chatty the protocol (i.e., the more round trip times required for communication), the lower the overall throughput. As a result they recommend the development of a bulk-mode of operation for chatty protocols such as HTTP. Consistent with Ott and Kutscher, they remark that the numerous authentication stages in 802.11 networks must be eliminated. Similarly, Zhuang et al. [76] point out that loss of control messages during application startup causes significant reduced throughput. In this thesis, we quantify the effects of the numerous stages of connection setup during an opportunistic connection and show that loss of control messages results in a significant reduction in the amount of data that is transferred. We specifically focus on the last hop of the connection, which is the wireless link from the access point to the vehicle. Therefore, in our experiments we do not use a backhaul connection to the Internet as we only transfer data between the access point and the vehicle. This is applicable to a realistic scenario because the backhaul link could be a bottleneck to the connection and therefore storage at the access point would be required to allow it to act as a cache between the vehicle and the Internet.

More recently, Bychkovsky et al. [10], as part of the CarTel project at MIT, conducted an extensive empirical analysis of the performance characteristics of using existing open/unsecured 802.11 networks for vehicular Internet access. Based on data they collected from nine vehicles under normal driving conditions in urban environments for almost one year, they found the median connection duration to be 13 seconds and the mean duration between connections to be 75 seconds, showing that existing 802.11 access points can be used by vehicular users for a variety of applications. They found that long connection setup times significantly reduce the useful length of the connection. Specifically, they found the mean scan, mean association, median IP acquisition, and median application initialization times to be 750 ms, 560 ms, 1.83 seconds, and 8 seconds, respectively. They propose an IP address caching scheme which by-passes DHCP and reduces the median IP acquisition time to 346 ms. They attribute their high application initialization time to overloaded client devices, due to running database software. Despite reporting these measurements, they do not identify causes of these delays, nor characterize their impact on the overall connection. They further remark, consistent with Ott et al. [49] and Gass et al. [18], that high losses at the beginning of a connection dramatically reduces the overall throughput of a connection and cite this as an area of future investigation.

The VanLan project at Microsoft Research has measured opportunistic communication between vans and WiFi access points in urban areas [42]. Their primary finding is that in urban areas the signal quality does not exhibit the same increasing, peak, and then decreasing pattern as found previously in line-of-sight highway scenarios [49, 27, 10]. In particular, in urban areas, signal quality drops sharply and unpredictably during an opportunistic connection [42]. They find that these *gray periods* do not occur consistently at the same location and are difficult to predict using available statistics such as RSSI and frame loss. They conclude that minimizing disruptions requires new protocols and that predictions based on past performance at a location can help identify gray periods. In this thesis we draw similar conclusions in a rural setting. We find that the signal quality of an opportunistic connection is very repeatable, as shown in Section 4.1.1. As well, in Section

4.5, we similarly conclude that an access point could exploit its knowledge of past connection history to help future connections.

To improve the interactive experience, particularly for VoIP calls, during an opportunistic connection, the VanLan project has developed a pseudo-handoff protocol named ViFi [5]. ViFi exploits access point diversity during opportunistic communication. The basic idea is that if an access point overhears a packet intended for another access point but does not overhear an ACK, the overhearing access point will attempt to deliver the packet itself. Using this protocol in their testbed doubled the length of disruption-free VoIP calls compared to an 802.11-like handoff protocol [5]. Despite improvements for interactive applications, they remark that for non-interactive applications, the choice of hand-off method is not critical. Their work is complementary to work in this thesis as we focus only on batch transfers occurring between mobile devices and a single access point.

If vehicles are equipped with steerable directional antennas, Navda et al. [47] demonstrate that the direction of the antenna can be set based on past history to significantly improve overall throughput.

Infostation Applications: Several novel uses of infostation-style connectivity have been studied. ZebraNet [37] was one of the first such projects, which involves attaching sensors to wild zebras that record movement and other data which is eventually relayed back to a collector node. Small et al. [61] studied a similar system for tracking the movement of whales. Such work can benefit from the experimental analysis of opportunistic connections done in this thesis.

Finally, a novel way of distributing common content to vehicles passing a roadside access point was proposed by Nandan et al. [46] which involves a BitTorrent [8] like approach to sharing common content. Using this technique, vehicles traveling along a highway would each receive different chunks of the same file as they passed the access point. Since they are likely to be in close proximity of each other for a much longer period of time then they are within range of the access point, the vehicles can share chunks with each other after they have left coverage of the access point. Understanding network protocol operation during communication between vehicles and between vehicles and roadside access points is essential to the success of such a proposed application. The analysis of network protocols in vehicle to roadside communication in this thesis provides a significant first step toward such understanding.

2.2.3 Ferry-Based

In ferry-based communication, a special mobile middle-node is employed to take on the responsibility of carrying data from source closer to the destination. Two general areas of message ferrying have been studied:

Ad-hoc Message Ferrying: In this paradigm, the mobility pattern of the nodes is not known and in order to aid in delivery of messages between partitioned areas of the ad-hoc network, message ferries are introduced. Zhao et al. [75] were the pioneers of this idea and describe it as introducing non-randomness in the mobility of nodes to overcome network partitions. Zhao et al. have studied algorithms for determining the optimal path a message ferry node should take in order to maximize the delivery-ratio in delay-tolerant networks.

Scheduled Message Ferrying: In this area of work, a rough schedule of the message ferry is known and the ferry is relied upon to be the main source of connectivity between distant nodes and the bulk of the network. The Tetherless Compting Architecture uses bus-based ferrying in their KioskNet project [59]. The primary goal of KioskNet is to provide basic Internet connectivity to rural villages in developing nations such as India. In villages with no other means of connectivity, a city bus with an 802.11 access point serves as the only means of connectivity between the kiosk in the village and the Internet. As a bus drives past a village kiosk, it is important to maximize the amount of data transferred, as is the goal of our work in this thesis.

2.2.4 Scheduled Access

A unique situation arises in a mostly disconnected network when the schedules of links are precisely known, such as in inter-planetary communication [64]. Because the schedules of links are known precisely, optimal routes can be computed and relied upon when performing routing in deep space delay-tolerant network [12]. This allows the accurate *a priori* computation of delivery ratio, waiting time, and quality of the network, which is not possible in general ad-hoc routing. Fall et al. [34] and Jones et al. [36] study scheduling in general delay-tolerant networks that assume links follow a precise schedule that can be learned. Similarly, Wang et al. [68] deal with the same set of problems in PostMANET which explores the use of the postal system to provide a mechanism for digital communication. Although schedules are precisely known, maximizing data transfer during an opportunistic connection is still dependant on network protocol behavior, which is explored in this thesis.

2.3 Wireless Performance Optimizations

Improving throughput, reducing delay, and increasing fairness have been the subject of much study in the context of wireless communication. We next discuss those studies related to opportunistic vehicular communication.

2.3.1 Optimizing TCP for Wireless

The problem of TCP's poor throughput over wireless links arises because TCP assumes all loses are due to congestion rather than link loss, as depicted previously in Figure 2.2. Four approaches have been taken to improve TCP's performance over wireless networks:

- Split the end-to-end TCP connection into two halves, one for the wireless side and one for the wired side. This was first proposed by Bakre et al. [1] and titled Indirect-TCP.
- Use a TCP-aware middle-box, such as a wireless access point, that detects TCP losses on the wireless side and generates local retransmissions. This idea was first proposed by Balakrishnan et al. [4] in Snoop.
- Modify TCP to be link-aware. This approach is taken by Explicit Loss Notification [2] and variants of TCP such as WTCP [60].
- Hide wireless loses by using automatic repeat requests (ARQ) and/or forward-error correction (FEC) at the link layer. This is done in 802.11 and virtually all cellular networks.

Balakrishnan et al. [3] compare a variety of these approaches and find that all achieve similar results, but each are better suited for different operating scenarios, including whether or not client modification is reasonable and what functionality is available on the access point.

The majority of this past work was done using wireless technology that did not perform link-layer retransmissions and before 802.11 was first released in 1997. 802.11 performs up to eight link-layer retransmissions, thus reducing the number of link-layer losses seen by TCP. In Section 4.1.3, we show that 802.11 MAC retransmissions are very effective and TCP sees only a small fraction of link-layer losses.

As a result of the aggressive 802.11 MAC retransmission scheme, when using TCP over 802.11, TCP is likely to see link-layer losses only during periods of poor signal quality, when all of the MAC retransmissions are likely to fail. During an opportunistic connection, when the vehicle first enters coverage of the access point, the signal is weak and link losses are common. This is a delicate point in the connection because TCP is more likely to see a loss and then backoff, thus reducing the useable amount of the connection. We show examples and discuss this further in Chapter 4.

2.3.2 802.11 Bit Rate Selection

When sending data, 802.11 devices must choose among a set of possible encoding rates to send each data frame: four rates in 802.11b, eight in 802.11a and 12 in 802.11g. Choosing a particular rate corresponds to using a different modulation scheme and error coding at the physical layer. Choosing an appropriate rate is challenging because only previous loss statistics and a rough estimate of channel energy are available.

Since rate selection algorithms are not part of the 802.11 standard, vendors are free to implement any scheme they desire. Research using 802.11 hardware has shown that such rate selection schemes are significantly different between cards and even between drivers [21], and furthermore that performance is very dependent on the environment [26, 71, 7].

Much research has been done on 802.11 rate selection algorithms, including the AMRR [39], ONOE [44], and SAMPLE [7] algorithms as well as the more recently proposed RRAA [71] algorithm. Research on a new bit rate selection algorithm for vehicular opportunistic communication is a fruitful area of future work. In this thesis, we show in Section 4.1.2 that the default parameters of the default bit rate selection algorithm (SAMPLE) used in the Madwifi drivers are inadequate for vehicular opportunistic communication because they rarely select rates above 11 Mbps even when 54 Mbps was possible. Modifying the default parameters was necessary in order to make use of higher rates, as discussed further in Chapter 4.

2.3.3 802.11 Performance Anomaly

When nearby 802.11 devices communicate simultaneous, the throughput of each device is reduced because the wireless medium must be shared between all devices in range of each other. During periods of overload, the 802.11 MAC protocol is designed such that, on average, all senders are able to send the same number of data frames. However, because each sender may choose different bit rates to use, it is possible for one user to dominate the time using the wireless medium by using a low bit rate; i.e., sending a MAC frame at 1 Mbps takes approximately 40 times longer than if the frame were sent at 54 Mbps. This results in choking out users capable of sending at high rates because 802.11 allows stations to send one frame at a time, in turns. As such, all users are effectively slowed down to the rate of the slowest user. This has been dubbed the 802.11 performance anomaly by Heusse et al. [28].

The most prominent solution to the performance anomaly problem is to fairly share the amount of time each user can access the medium rather than each user being allowed to transmit an equal number of frames. The notion of time fairness, proposed using different methods of implementation [63, 73, 45], results in each user accessing the medium for an equal amount of time, but because users can choose different encoding rates, each may achieve a different data rate.

The problem of how to fairly allocate access to the wireless medium generalizes into a scheduling problem of how to optimally schedule a shared resource. In the case of vehicles driving along a highway, opportunistically communicating with 802.11 access points, it was shown by Hadaller et al. [25] that in the presence of multiple vehicles, simply allocating access to the channel to the vehicle with the strongest signal results in significant throughput gains over both standard 802.11 and time fairness, while maintaining a high degree of fairness between vehicles.

Although there has been much related work involving vehicular communication, none examines detailed data-link and transport layer protocol interaction as done in this thesis. Our experimental analysis complements existing work by exploring vehicular opportunistic communication at a level of detail not previously explored. We next discuss the experimental setup we used for our analysis.

Chapter 3

Experimental Setup

This chapter details the experimental hardware and procedures used to analyze protocol behavior during opportunistic communication, as well as the steps taken to account for systematic and random error in GPS measurements. Our goal is to examine the detailed protocol interaction during opportunistic communication involving a vehicle driving past an 802.11 roadside access point at highway speeds.

3.1 Equipment

All experimental runs were done with a 1997 Saturn SC-2 traveling at 80 km/h, the speed limit on the rural road. The vehicle drove past our roadside access point that was setup on the top of a five foot (1.52 m) step ladder. We used two dedicated sniffers, one situated beside the access point and one on the car, depicted in Figure 3.1.

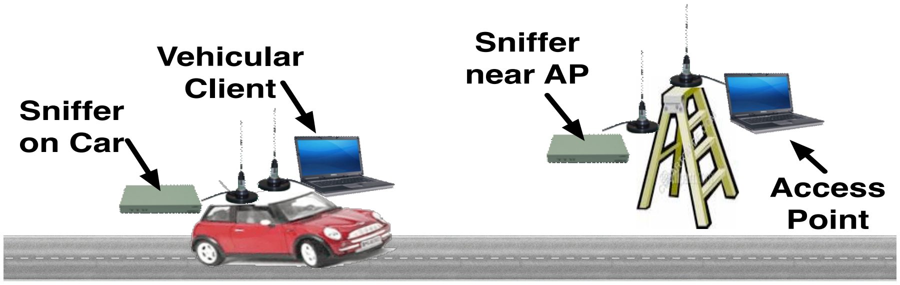


Figure 3.1: Experimental setup used to analyze opportunistic vehicular connections using 802.11 hardware.

We used the following equipment, pictured in Figure 3.2:

- **Access point**: Dell Latitude CPX H500GT laptop with 500 MHz processor and 512 MB RAM, with magnetic GlobalSat BU-353 USB GPS receiver, Atheros-based CB9-GP-EXT CardBus 802.11 a/b/g wireless card and 7 dBi

Pacific Wireless MA24-7N magnetic-mount external omnidirectional antenna placed on a five foot (1.52 m) step ladder.

- **Vehicular client**: Dell D600 laptop with 1.6 GHz processor and 1 GB RAM, with the same BU-353 GPS receiver, CB9-GP-EXT wireless card, and MA24-7N antenna. The GPS receiver and wireless antenna were mounted externally on the roof of the vehicle as shown in Figure 3.2.

- **2 dedicated sniffers**: Each is a Soekris net4801 single-board computer with 266 MHz 586 class processor, 256 MB RAM, 40 GB hard drive, with Atheros-based EMP-8602 miniPCI 802.11a/b/g wireless card and the same MA24-7N antenna. One sniffer was located in the vehicle with its antenna mounted on the roof the vehicle, the other located on the step ladder near the access point.

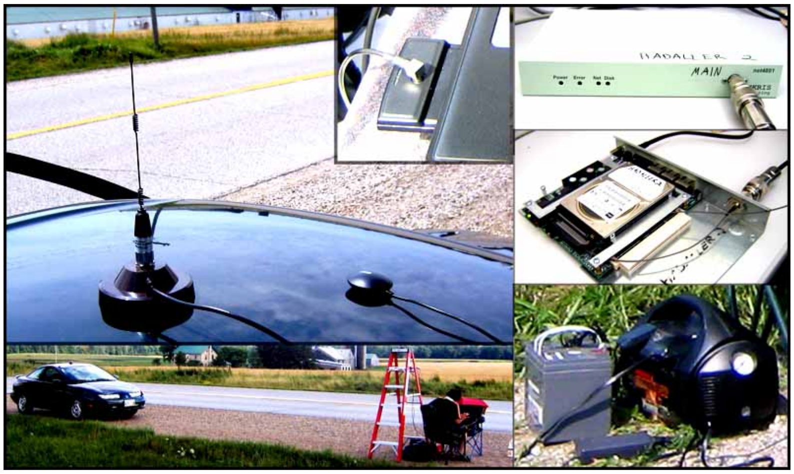


Figure 3.2: Equipment: Soekris board (top right), 802.11 PCMCIA card in laptop with external antenna connector (middle top), Magnetic 7 dBi Antenna and Magnetic USB GPS Receiver (left), Vehicle and roadside access point (laptop with antenna on ladder) (bottom left), and the two power units used to power the access point (bottom right).

All hardware ran the latest release of the Debian Linux version 3.1 testing branch at the time (August 2006) with kernel version 2.6.16 and Madwifi driver version 0.9.1 [44].

We report on a total of 48 runs in this thesis. 15 of those were used to test the duration of the connection and the remaining 33 focused on the connection setup phase.

Although only one hardware configuration is used in our study, we discuss how our results can be generalized in Section 5.1.

Sniffer Configuration: We used a total of four sniffing devices: sniffing software running in the background on the access point and the client, as well as two dedicated sniffers, depicted in Figure 3.1. This arrangement ensures that, with high probability, all packets sent on the air were captured at one of the sniffers. Our data set unifies these four captured traces.

3.2 Operating Parameters

In order to focus on protocol operation, we fixed the following variables: all runs were performed at 80 km/h, during the same day, using downstream TCP traffic with the access point initiating a TCP connection to the vehicle. Downstream TCP traffic was used because it represents the expected flow of the majority of content to a vehicle in a realistic environment [57]. Although the vehicle will likely initiate the TCP connection in practice, in our experiments, the TCP sender (the AP) had to initiate the connection due to a limitation of our traffic generation software. We argue that similar results would have been obtained had the vehicle initiated the TCP connection or if data were sent in the uplink direction because the underlying causes of decreased overall throughput remain unchanged. Furthermore, vehicular experiments not reported in this thesis indicate that weather conditions do not significantly impact signal quality between the access point and the vehicle. As well, we argue that our results can be extrapolated to different vehicle speeds, as discussed in Section 5.1.

We chose to experiment with TCP as it is the transport protocol used by the majority of existing applications. However, in ongoing work, we have experimented with UDP and preliminary results show very little difference between UDP and TCP due to the aggressive retransmissions done at the MAC layer, as discussed further in Section 4.1.3.

Although we aim for Internet access in vehicles, in our experiments we only consider the last hop of the connection, the wireless link from the access point to the vehicular client. We ignore the effects of the backhaul connection to the Internet because we assume that the access point has some form of local storage and can act as a cache between the vehicular client and the Internet. This is necessary because 802.11 data rates are much higher than a typical cable or DSL backhaul connection and without local storage the backhaul would be a bottleneck.

We used statically configured IP addresses, as DHCP is known to behave poorly in this environment [10], and we were interested in isolating less well-studied protocol behavior. We used the default transmit powers in the Madwifi driver of 19 dBm and 15 dBm for the access point and the client, respectively.

All experiments were conducted on the same section of road, a straight, relatively flat (some slight inclines and bumps were present), undivided country road surrounded by tall corn crops on one side and power lines and the occasional house on the other. Other vehicular traffic was light or non-existent during our experiments. Exploring different environments is an area of future work, as discussed in Section 5.1.

During our initial experiments, we found that the default parameters of the default bit rate selection algorithm (SAMPLE) used in the Madwifi drivers, were not responsive enough for our environment, and rarely selected rates higher than 11 Mbps. Modifying these parameters was necessary in order to make the algorithm more responsive for the vehicular environment. The details of these modifications are discussed in Section 4.1.2.

3.3 Logging

All transmissions between the access point and client were captured by placing the Atheros card in monitor mode and using tcpdump version 3.9.4 to capture all frames, including extra MAC layer information from the card in the Prism monitoring header, such as the MAC bit rate and measured RSSI for each frame.

We used GPS devices, attached via USB, to record the position of both the access point and the vehicle over time. We used a shell script loop to poll the GPS device for its position once per second, which is the highest frequency of measurements supported by the GPS device. Because of the high accuracy that was required, it was necessary to correct for certain systematic GPS errors, discussed in Section 3.5.

Time synchronization between all devices was performed using NTP between experimental runs.

3.4 Experimental Procedure

Our experiments were conducted as follows. The vehicle begins out of range of the access point and the logging scripts on the access point, the vehicle, and both sniffers are started. The vehicle, driving at a constant speed of 80 km/h, enters the range of the access point and continues driving at a constant speed until it leaves coverage range. The experiment is then repeated in the opposite direction.

We used iperf v1.7.0 [33] to send bulk TCP data from the access point to the client. At the start of an experiment, the vehicular client runs iperf in listener mode, waiting for a connection from the iperf sender. Once the client enters the range of the access point, it performs a standard 802.11 MAC association with the access point. Using a shell script, the access point detects a newly associated client and launches the iperf sender, which initiates a bulk TCP connection to the statically configured client IP.

3.5 Achieving Relative GPS Accuracy

We rely on consistent GPS measurements between different experimental runs and accurate relative measurements between GPS devices to determine the vehicle's precise relative position with respect to the access point. This is not the same as achieving absolute GPS accuracy, that is, ensuring that the measured GPS position is close to that of the actual ground position.

In this section, we analyze GPS measurements and show that up to 49.9 m of position error is present and can be reduced to less than 8.6 m after accounting for systematic error. We next present some brief background information about GPS.

3.5.1 Understanding GPS

The Global Positioning System (GPS) is a U.S.-operated system for determining terrestrial position using signals from satellites. The 31 GPS satellites follow a Medium Earth Orbit at 20,200 km above the Earth and orbit the planet twice each day, traveling at over 14,000 km/h [22]. Satellites each transmit a unique repeating code over the 1.5 GHz carrier along with their position. These signals are used by a GPS receiver to determine precise distance to each satellite. Signals from at least three different satellites are used to determine a 2D position fix, and at least four satellites for a 3D fix.

The position measured by the receiver is subject to error from many sources [41]. Because the atmosphere is not a vacuum, effects of the dense ionosphere introduce non-uniform delay in the signal sent from the satellite. In addition, because satellites drift slightly off course, their reported positions can be inaccurate enough to cause measurement errors on the ground. Multi-path and shadowing effects as well as imprecision in the receiver's oscillator can also reduce accuracy.

3.5.2 Previous GPS Measurements

Quantifying GPS error has been studied previously. Wing et al. [70] found stationary accuracy of commodity devices to be within 5 m to 10 m of true position depending on the view of the sky. Webster et al. [69] found 3-6% error in measuring the size of a 500 m field. A study by the U.S. National Geodetic Survey [67] found 95% of GPS measurements fell within a radius of 6.3 m. In addition, the manufacturer of the devices we used in our experiments claims a 10 m position accuracy [19]. Although the absolute accuracy of stationary GPS measurements has been examined by other researchers, none have measured relative accuracy between devices or the impact of mobility as we do in this thesis.

3.5.3 Stationary Experiments

In addition to data gathered from the vehicular experiments, we also performed stationary experiments to measure consistency between GPS devices. In these experiments, four identical devices were mounted within 10 cm of each other, as shown in Figure 3.3. These devices were same GlobalSat GPS receivers that were

Systematic Error				
Error	**Impact**	**Mitigating Error**	**Sec.**	**Fig.**
Infrequent Measurements	Up to 0.5 s, or 11.1 m at 80 km/h	Performed linear interpolation between measurements.	3.5.4	3.4
Measurement Delay	1.36 s ± 0.06 s (95% CI), or 30.2 m ± 0.4 m at 80 km/h	Subtracted 1.36 s from measurement timestamp.	3.5.4	3.5, 3.6
Random Error				
Error	**Impact**	**Mitigating Error**	**Sec.**	**Fig.**
Single-Device Variation	$\leq$ 4.3 m in 95% of measurements.	Averaged stationary measurements.	3.5.5	3.7
Device Consistency	No significant inconsistency, error captured by single-device variation.	Cannot be corrected.	3.5.5	3.8, 3.9

Table 3.1: Sources of GPS Error

used in the vehicular experiments. The devices were magnetically mounted on a metal sewer grate in the middle of the University of Waterloo campus. The experiments were performed on a day with no clouds, and all devices had a clear view of the sky.

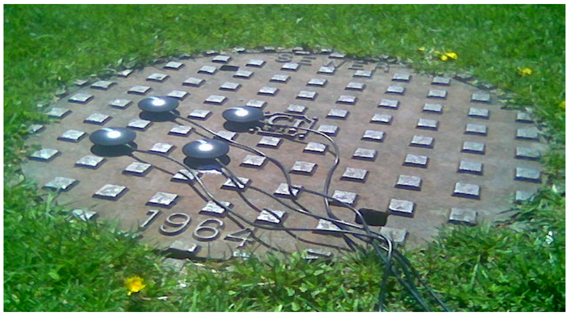


Figure 3.3: Arrangement of GPS devices used for Stationary Measurements.

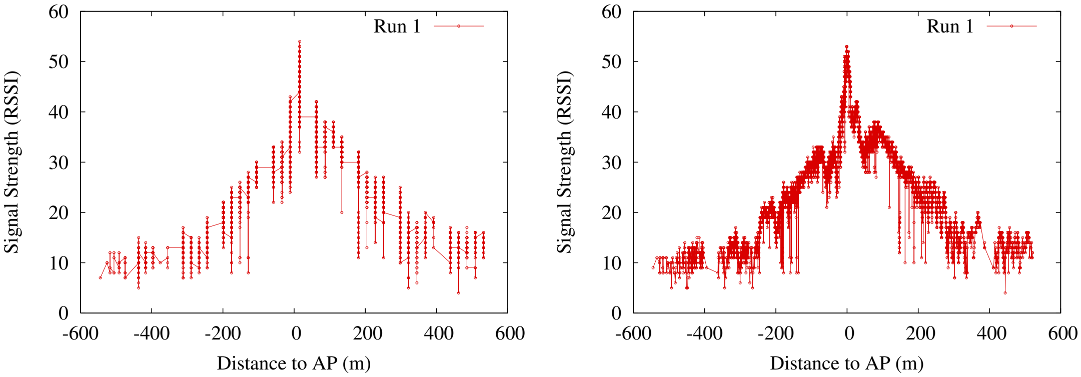
Table 3.1 summarizes the sources of GPS error we identified. We next discuss systematic error, followed by random error.

3.5.4 Systematic Error

Infrequent Measurements

Our off-the-shelf GPS receiver only reported position measurements once per second, which is common in commodity GPS receivers [13]. This is problematic when trying to determine precise position over short time periods.

In our vehicular experiments there were thousands of frames transmitted per second. Simply assigning a frame's location to the nearest GPS measurement results in up to 0.5 s of error, which translates into 11.1 m at 80 km/h. Using this method results in a sparse signal strength map, shown in Figure 3.4(a).



(a) Raw Measurements

(b) Interpolated Measurements

Figure 3.4: (a) Using the once-per-second position measurements from GPS results in a sparse signal strength map and as much as 0.5 s of error, or 11.1 m of error at 80 km/h. (b) Performing simple linear interpolation between position measurements improves accuracy and smooths the map.

Correcting for Infrequent Measurements: We performed linear interpolation between measurements. A linear fit was appropriate because we were traveling at a constant speed and did not change direction. In situations where the mobile device changes speed and/or direction, a more complex interpolation would need to be performed. Figure 3.4(b) shows that using interpolation results in a much smoother signal strength map and more accurate frame locations.

Measurement Delay

In this section we describe how we determined the amount of GPS measurement delay. We conclude that when a position was received on the USB port it was actually the position of the vehicle 1.36 seconds ± 0.06 s (95% confidence interval) in the past. Thus, at 80 km/h, the vehicle was 30.2 m ± 0.4 m ahead of the reported position at any point in time.

Determining GPS Measurement Delay: We were able to determine the actual position of the vehicle relative to the access point based on the measured signal strength of the wireless signal. The strength of a signal received by a wireless device is proportional to the inverse of the distance to the sender [56]. Therefore, the signal strength will be at its maximum when the vehicle is closest to the access point. Our signal strength measurements exhibited a sharp peak, lasting an average of only 62 ms ± 28 ms (95% CI) (1.4 m ± 0.6 m at 80 km/h), as can be seen in Figure 3.4(b). It is therefore reasonable to assume that the center of the signal strength peak is the position where the vehicle was closest to the access point. With this point of reference, we can determine the GPS measurement error as follows.

Figure 3.5(a) shows the position of the GPS measurement before the access point (x_{gps1}), at the access point (x_{ap}), at the signal peak (x_{peak}), and the GPS measurement after the access point (x_{gps2}). Recall that the GPS device only reports measurements once per second, thus interpolation is needed. After a GPS measurement is taken, it is reported by the device and read by the host system on the USB port, this takes Δ_{gps}^{read} time. The time to attach a timestamp and log this measurement is Δ_{gps}^{write}. Similarly, the time to read a signal measurement is Δ_{signal}^{read} and to log it takes Δ_{signal}^{write}. The access point's position is known as it has its own GPS device[1].

The signal peak is assumed to occur when the vehicle is closest to the access point ($x_{ap} = x_{peak}$). Therefore, the GPS measurement delay (Δ_{gps}^{read}) is the difference between the time that the vehicle was measured to be closest to the access point ($\widetilde{t}_{gps\_peak}$, requires interpolation) and the time that the signal peak actually occurred ($\widetilde{t}_{peak} - \Delta_{signal}^{read} - \Delta_{signal}^{write}$), as follows:

We first determine the interpolated GPS timestamp at x_{peak}:

$$\text{Define } \widetilde{t}_{gps\_peak} = \widetilde{t}_{gps1} + \frac{x_{peak} - x_{gps1}}{\text{velocity}}$$

Because $\widetilde{t}_{peak}$ and $\widetilde{t}_{gps\_peak}$ are both relative to x_{peak}:

$$\begin{aligned} \widetilde{t}_{gps\_peak} - \Delta_{gps}^{read} - \Delta_{gps}^{write} &= \widetilde{t}_{peak} - \Delta_{signal}^{read} - \Delta_{signal}^{write} \\ \Delta_{gps}^{read} &= \widetilde{t}_{gps\_peak} - \widetilde{t}_{peak} + \Delta_{signal}^{read} + \Delta_{signal}^{write} - \Delta_{gps}^{write} \end{aligned}$$

We make two assumptions in order to isolate GPS measurement delay (Δ_{gps}^{read}), these do not affect the amount of error experienced: (a) the time to log a GPS measurement and log a signal strength measurement are, for practical purposes, identical and negligible ($\Delta_{gps}^{write} = \Delta_{signal}^{write} = 0$), and (b) the time to report a signal strength measurement is negligible ($\Delta_{signal}^{read} = 0$), yielding a best case for Δ_{gps}^{read}:

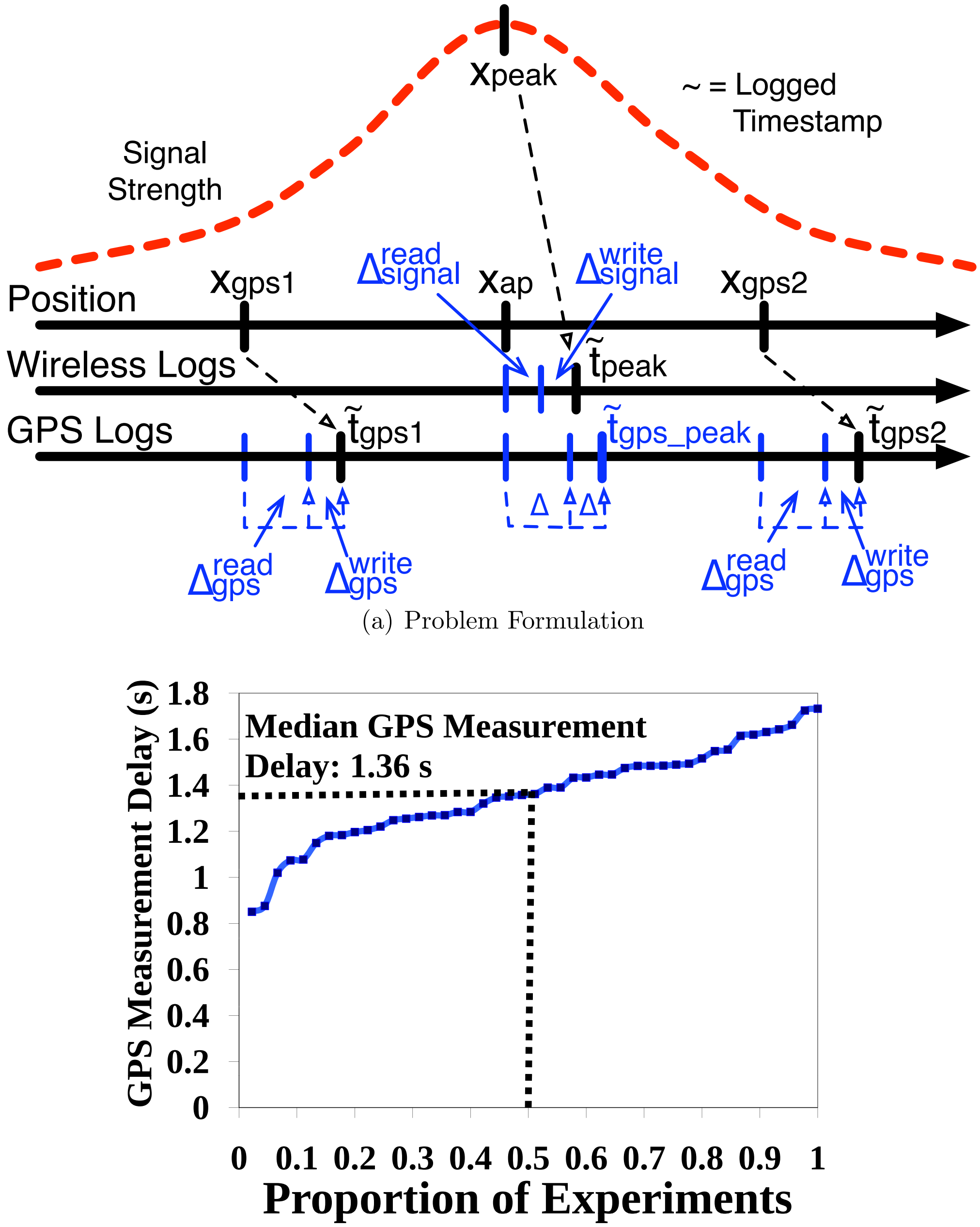
$$\text{GPS measurement delay } \Delta_{gps}^{read} = \widetilde{t}_{gps\_peak} - \widetilde{t}_{peak} \tag{3.1}$$

We calculated the GPS measurement delay using Equation 3.1 from logs of 45 mobile experiments. Figure 3.5(b) shows a CDF of Δ_{gps}^{read}; the average delay was 1.36 s ± 0.06 s (95% CI).

The cause of GPS measurement delay is primarily due to the sequential process of measuring signals from each satellite, computing the receiver's position, and then sending the data to the host laptop [13]. As a result, when a computation is received by the host laptop, the measurement is already out-of-date.

Correcting for Measurement Delay: Subtracting 1.36 s from the timestamp reported by each GPS measurement was adequate to correct for GPS measurement

[1] Due to stationary variation, we averaged the measured positions of the access point.



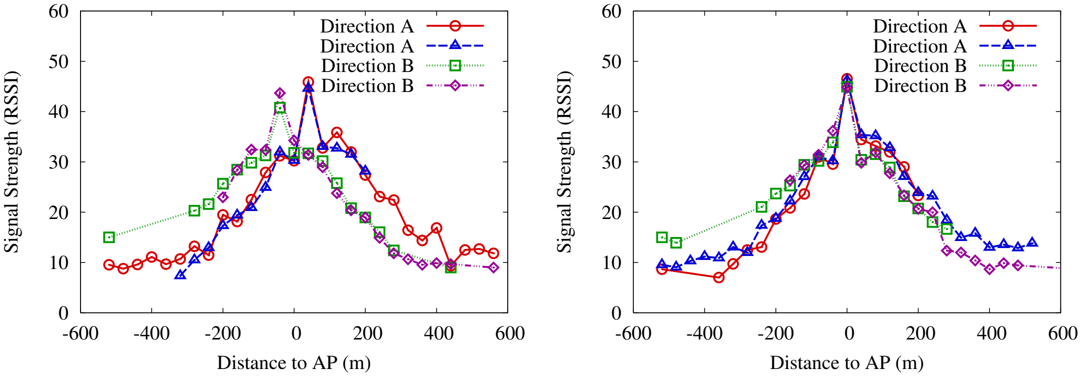
(a) Problem Formulation

(b) Empirical Measurements

Figure 3.5: Figure (a) formulates the problem of determining GPS measurement delay Δ_{gps}^{read}. Figure (b) shows the CDF of Δ_{gps}^{read}, calculated from 45 mobile experiments.

delay in our experiments. We caution the reader that other devices may experience different delay, and the measurement delay should be determined before applying any correction. However, we expect that the majority of off-the-shelf devices will experience similar delay as it is inherent to the operation of standard NMEA GPS devices.

Validation of Correction for Measurement Delay: We validate our correction by comparing experimental runs done with the vehicle traveling in the both directions on the road. Before correction, the signal peaks do not align (Figure 3.6(a)), and after applying our correction, the signal peaks align (Figure 3.6(b)). Therefore, our correction reduces error due to measurement delay.



(a) Before Correction

(b) After Correction

Figure 3.6: Validation of computed measurement delay; subtracting 1.36 s from the GPS timestamp corrects for GPS measurement delay. Two runs in each direction are shown, Figure A shows the difference in measured signal peaks before correction; Figure B shows that after correcting for this systematic error, the signal peaks align.

3.5.5 Random Error

Single-Device Variation

Over a 5.5 hour period, the measured position of the stationary access point varied by as much as 10 m. Figure 3.7 shows all of the measurements, along with the mean, median, and mode. Additionally, a box containing 95% of the measurements closest to the average is shown, of size 4.6 m × 5.7 m. Although we did not know the true location of the access point, the average of all stationary measurements is a reasonable approximation. Therefore, the error in 95% of measurements will be no more than 4.3 m, the distance from the average to the furthest corner of the box.

Reducing Impact of Single-Device Variation: Measurements can be averaged to reduce error due to stationary variation.

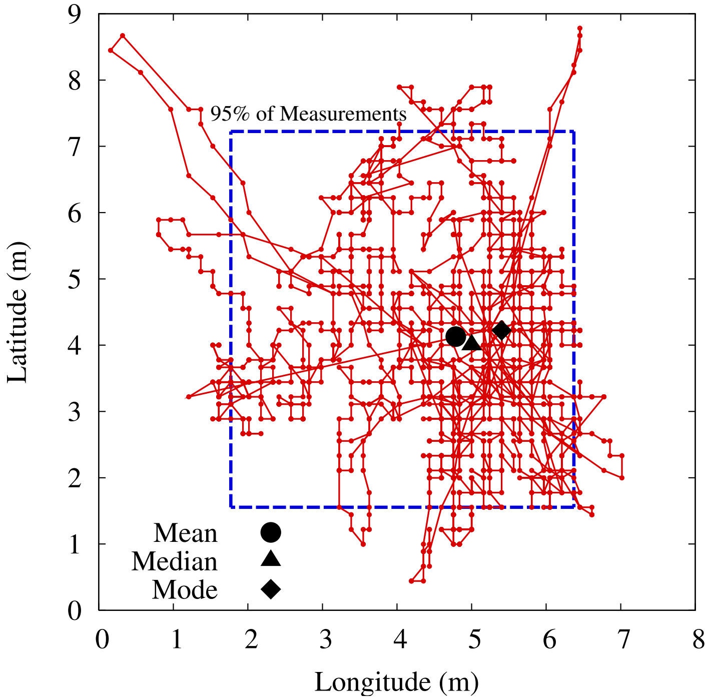


Figure 3.7: The measured position of the stationary access point over a 5.5 hour period.

Consistency Between Devices

For our vehicular experiments, we rely on accurate relative position measurements to determine the distance between the vehicle and the access point throughout an experimental run. If individual GPS devices exhibit significant measurement bias and produce significantly inconsistent measurements between devices, then determining accurate relative position would not be possible.

To investigate device-specific measurement bias, we placed four identical GPS devices within 10 cm of each other, as shown earlier in Figure 3.3. Upon first inspection some measurement bias appeared, as shown in the left half of Figure 3.8, showing 1800 measurements per device collected over a 30 minute period. However, 10 minutes later, measured positions from the devices all overlapped, shown in the right half of Figure 3.8. This indicates there is no prominent device-specific bias for our hardware.

However, at any point in time, there was a different of up to 10 m between measurements from the four devices. Further analysis revealed that this inconsistency was due to single-device variation and resulted from satellite selection done independently at each device.

Understanding Satellite Selection: During the 30 minute period shown in the left half of Figure 3.8, 15 satellites were detected by each receiver, eight were used the entire time, four were never used, and three were only used part of the time. Each

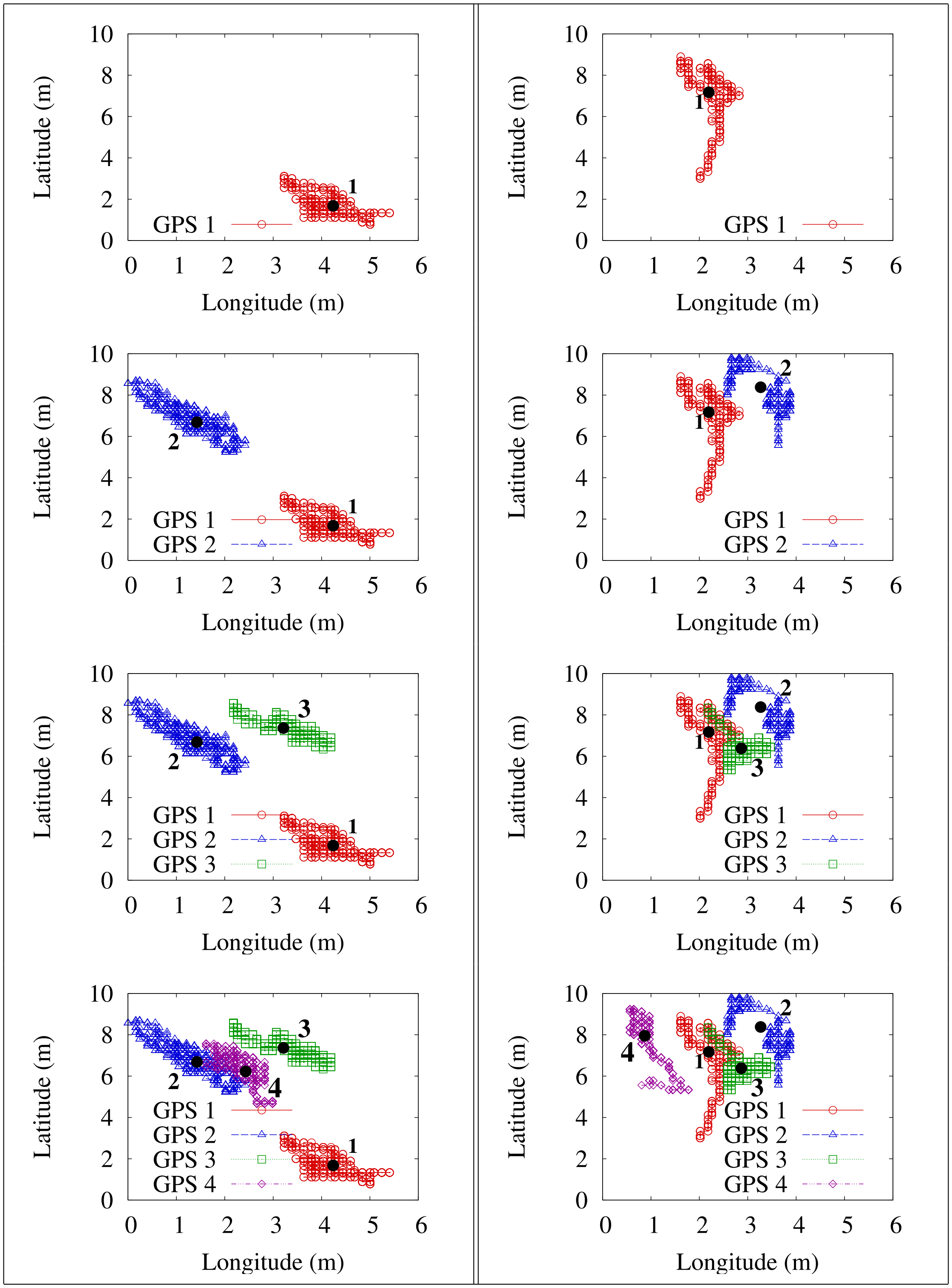


Figure 3.8: Left four (read top to bottom): Measurements from four nearby GPS devices over 30 minutes. Right four: The same four devices 10 minutes later, measured for 15 minutes. Average measured location is shown for each device.

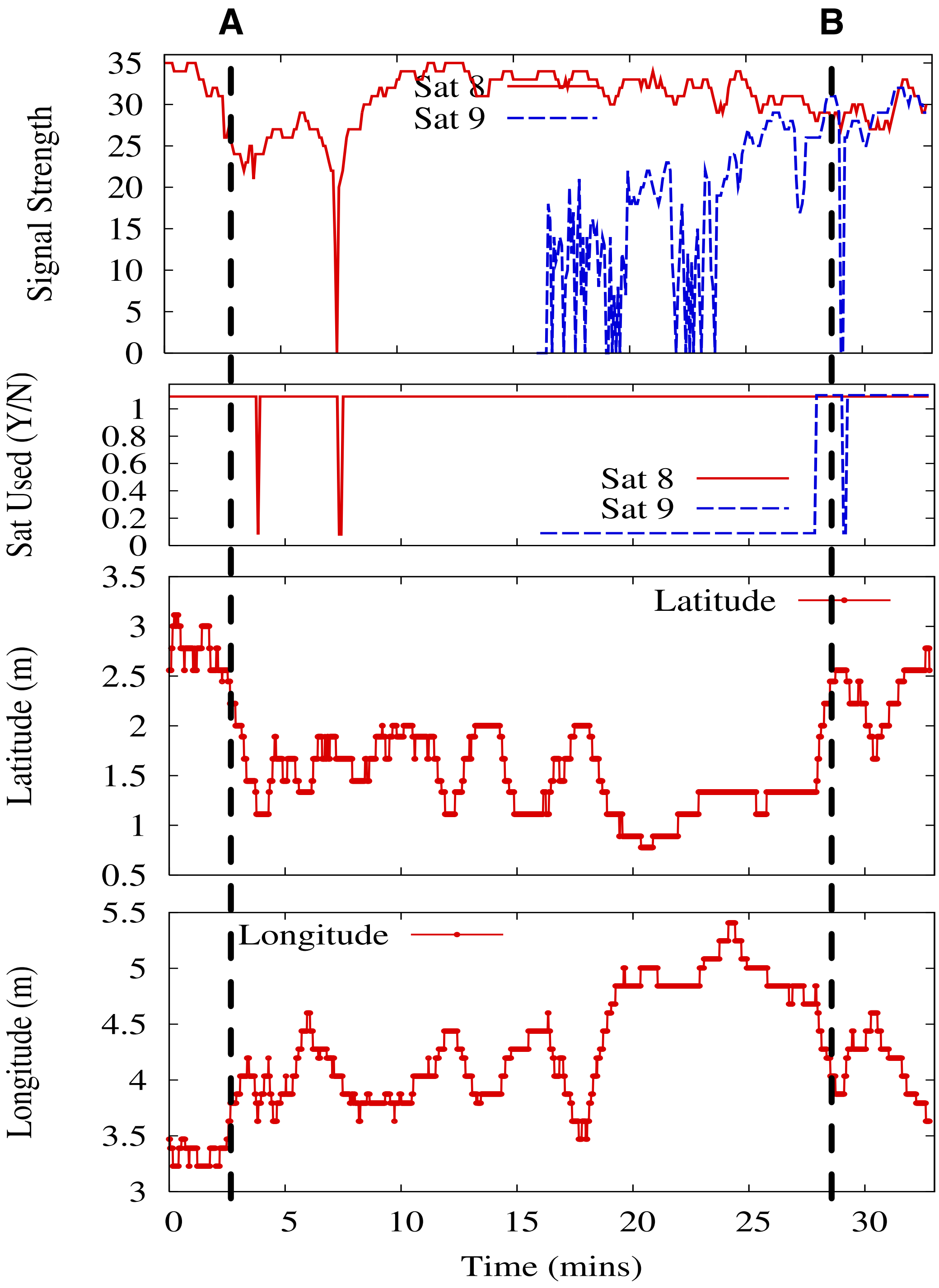


Figure 3.9: The effects of satellite selection over 30 minutes for a stationary device (GPS1, shown in the top left of Figure 3.8). At point A, satellite 8 was temporarily removed due to its weaker signal strength; at point B, satellite 9 was added. These changes affected both the latitude and longitude.

of the four devices selected a different subset of these three satellites at different times, leading to the reported position differences. To better understand the effects of satellite selection, Figure 3.9 shows satellite choices for a single stationary device (GPS1 from the top left of Figure 3.8).

We draw two conclusions from these measurements: (1) The strength of the signal received from a satellite determines whether the GPS receiver will use that satellite in the position calculation, and (2) changing which satellites are included in the calculation affects the measured position. Although satellite selection is only one of many causes of stationary error, it is important to understand its impact on measured position.

Reducing Inconsistency: This error cannot be reduced as the selection of satellites cannot be controlled with commodity hardware.

3.5.6 Summary of Improving Relative GPS Accuracy

Although random error cannot be eliminated, we have identified a significant amount of systematic error that can be corrected.

Without accounting for systematic error, up to 49.9 m of GPS measurement error is present; up to 41.4 m due to movement at 80 km/h and up to 8.6 m of random error (4.3 m at both the access point and the client). Performing linear interpolation between measurements and subtracting 1.36 s from the measurement timestamp corrects for systematic error due to infrequent measurements and measurement delay, respectively. Averaging the position measurements at the stationary access point will help reduce outlier effects due to random error but because the average is not necessarily more accurate than any other measurement, we do not consider this to have accounted for single-device variation. In addition, the random error at the vehicular client cannot be reduced as averaging cannot be performed. No significant inconsistency between devices was found beyond that captured by single-device variation.

Therefore, after reducing systematic error due to mobility, we are left with 4.3 m of error at both the access point and the client, resulting in a relative position error of less than 8.6 m. This amount of error is acceptable in our analysis, as we divide the roadway into 20 m intervals, as discussed in the following chapter.

Chapter 4

Results

4.1 General Observations

While conducting our vehicular experiments, we encountered several phenomena that led to further investigation. In particular we observed that: (1) connection quality is repeatable across runs, (2) the default MAC rate selection algorithm in the Madwifi driver significantly underutilized available connection quality, and (3) there were few TCP losses.

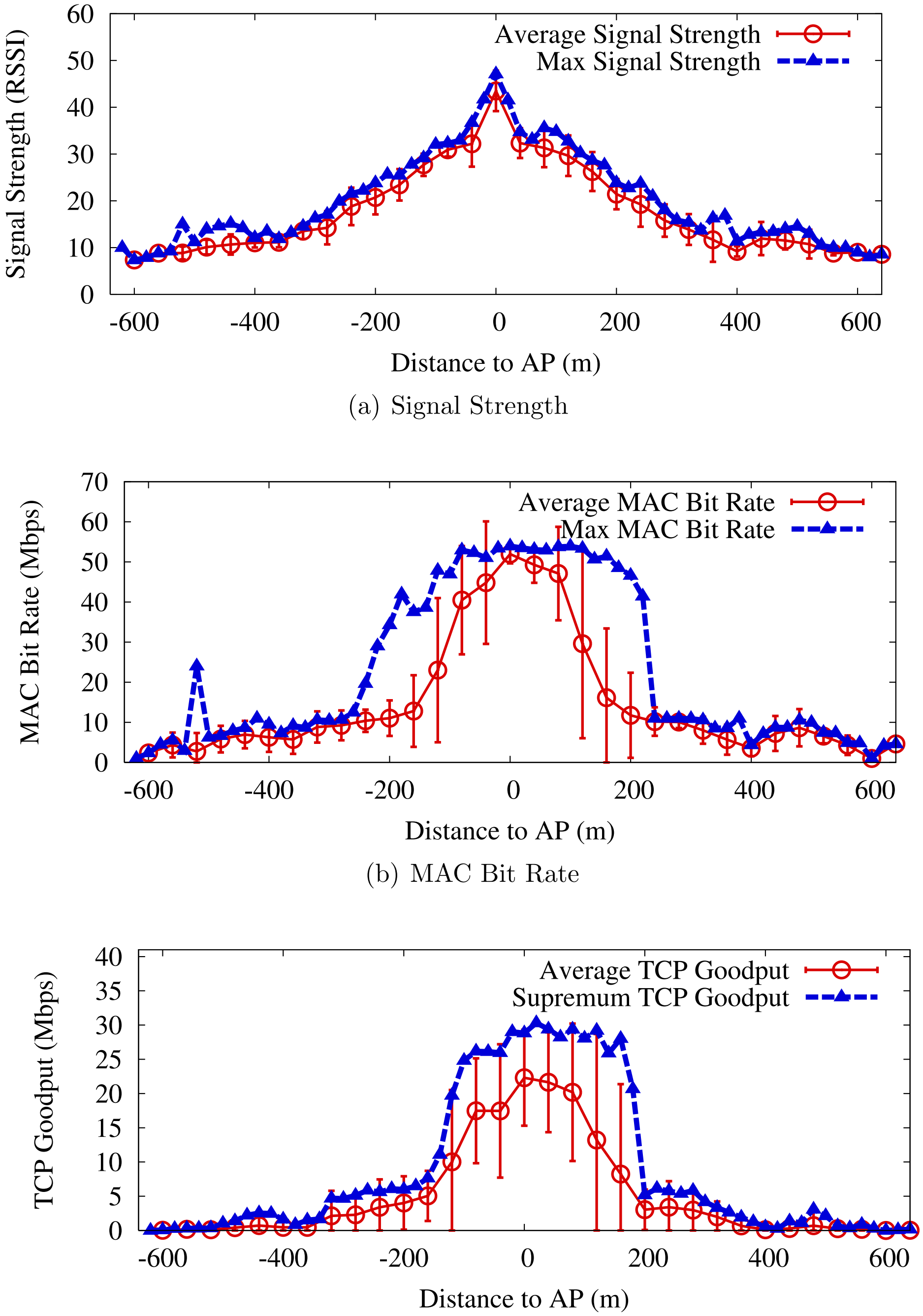
4.1.1 Repeatable Connection

We found that signal strength measurements, relative to position on the road, were very consistent between runs, as shown in Figure 4.1(a). Although the quality of the connection was repeatable, the behavior of data-link and transport layer protocols was very inconsistent, evident from the large 95% confidence intervals in Figures 4.1(b) and 4.1(c).

Because the connection quality was consistent, one would expect that the amount of data transferred between runs would also be consistent. Understanding why this is not the case is the focus of our experimental analysis in this chapter.

4.1.2 Default Bit Rate Selection Unsuitable

During our initial experiments, we found that the default SAMPLE bit rate selection algorithm [7] used in the Madwifi driver was not responsive enough for our environment, and rarely selected rates higher than 11 Mbps, resulting in significantly less data transferred. Figure 4.2 shows an example of the bit rates chosen by the default bit rate algorithm. For the same location, bit rates of up to 54 Mbps were possible, as shown in Figure 4.1(b).



(a) Signal Strength

(b) MAC Bit Rate

(c) TCP Goodput

Figure 4.1: Measured signal strength (RSSI), MAC bit rate, and goodput averaged over distance across 15 runs with 95% confidence intervals. The maximum potential connection spans a distance of 1280 m which lasted 58 seconds at a speed of 80 km/h.

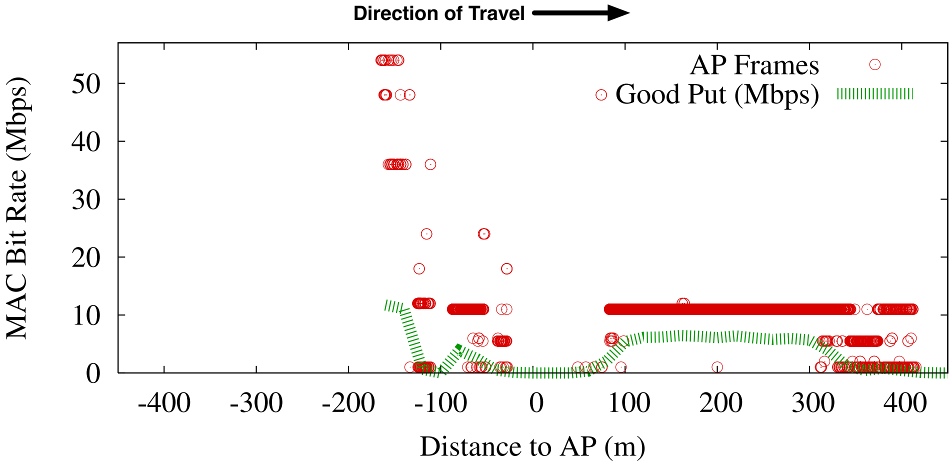


Figure 4.2: Example of poor rate selection using the default parameters of the default bit rate selection algorithm. Using these parameters would have resulted in significantly less overall throughput compared to using our modified parameters which achieved much higher rates (shown in Figure 4.1(b)). The lack of throughput during the middle of this run was due to a TCP timeout, caused by an initially excessive bit rate of 54 Mbps. The high initial bit rate also preventing the connection from starting until 150 meters before the access point.

In order to make use of higher bit rates we had to modify the default parameters of the default bit rate algorithm to make it more responsive to the rapidly changing environmental conditions. The default SAMPLE parameters were designed for stationary scenarios, and as a result, the algorithm considers frame loss statistics over the last 10 seconds when making a rate selection decision. Using a 10 second window is far too long for a vehicular opportunistic connection as significant signal quality changes occur on the order of a few seconds. Furthermore, the default SAMPLE parameters also only used 10% of frames to probe for new data rates and made a rate decision only every 1000 milliseconds. Again, in a scenario where the signal changes rapidly, more probes and more frequent decisions are necessary. We found that a window of one second, using 40% of frames to probe new rates, and making rate decisions every 100 milliseconds allowed SAMPLE to adequately choose bit rates during the vehicular connection. Our modifications are summarized in Table 4.1. We do not claim to have set the SAMPLE algorithm parameters optimally, merely that the default settings were unusably bad, and our settings are good enough to allow us to explore the problem further. Had we used the default parameters, TCP goodput would have been significantly reduced as the default parameters rarely chose rates above 11 Mbps. Optimal choice of MAC bit rate for vehicular scenarios is a fruitful area for future work.

Our modified bit rate selection parameters were used on the access point, which was the TCP sender in our experiments. However, we elected not to change the TCP client on the vehicle, as we wish to focus our analysis on what could be attained by an infrastructure provider without client modification. We experimentally explore

Parameter	**Original** [44]	**New Value**
Probe Packets	10%	40%
Sample Window	10 s	1 s
Decision Interval	Every 1000 ms	Every 100 ms

Table 4.1: Changes to Default SAMPLE Rate Selection Algorithm

the effects of using an unmodified client in Section 4.3.

4.1.3 Few TCP Losses

Although the number of MAC losses was quite high (18.5% of all transmitted MAC frames were lost), as shown in Figure 4.3, the 802.11 MAC retransmission scheme did an exceptional job of hiding losses from TCP. Of the approximately 22,000+ TCP packets transmitted on average per experimental run, there were only an average of 26.8 ± 33.3 (95% CI) losses seen by TCP (< 0.13% TCP loss).

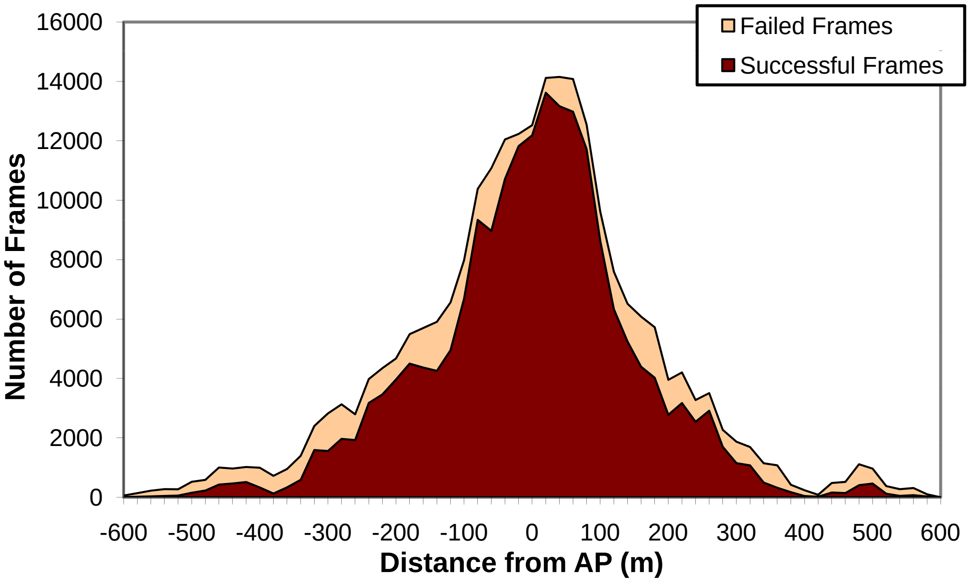


Figure 4.3: MAC retransmissions vs distance from the AP, summed across 15 runs. A higher proportion of losses occurred in the fringe areas and, on average, 18.5% of transmitted MAC frames were lost.

The reason for low TCP loss is the aggressive multi-rate retransmission scheme implemented in the Atheros Hardware Abstraction Layer (HAL). When the driver makes a call to the HAL to send a frame, it can specify up to four different rates to send the frame at, which the hardware will attempt in decreasing order, each with a maximum retry count of eight. Analysis of our logs revealed that frames were often transmitted 4 to 8 times at a high rate, then up to 8 times at the lowest rate, which would then usually succeed.

The overall number of TCP losses was very low. However, as explained in Section 2.3.1, during periods of weak signal such as at the start or end of an opportunistic connection, TCP is more likely to see losses due to the higher likelihood that all of the MAC retransmissions will be lost. We give some examples and discuss this further in Section 4.3.

4.2 Underutilized Vehicular Connections

In this chapter, we show that using existing data-link and transport layer protocols for opportunistic connectivity to vehicles results in (1) significant variation in the amount of data transferred per run as well as (2) significant underutilization of the connection. Figure 4.4 shows that the least data transferred in a run was less than half (42.3%) as much as the most data transferred in a run (21.6 MB vs 51.1 MB). Both of these are significantly distant from the median data transferred (32.6 MB), and all runs were below the potential, explained next.

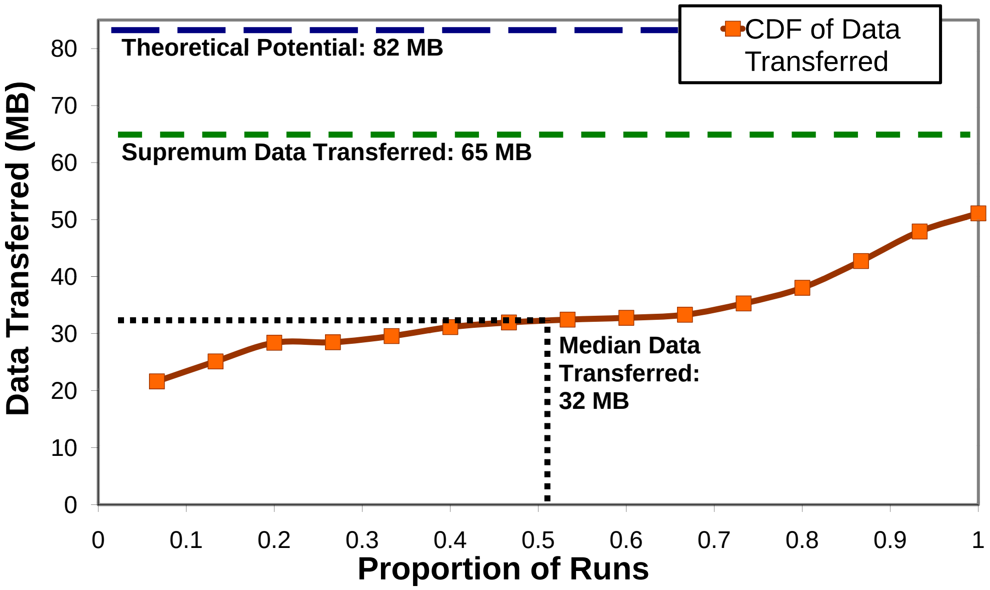


Figure 4.4: CDF of the amount of data transferred per run

Supremum: Because no run was problem-free, Figure 4.4 also shows the *supremum data transferred* of all runs, calculated as follows. Goodput for each run is computed over 20 m intervals (or sections of the roadway). The *supremum goodput* for each interval is the maximum goodput of the set of goodputs achieved in that interval over all runs (shown in Figure 4.1(c)). The supremum data transferred is then simply the sum of the supremum goodput of all intervals multiplied by the time spent in an interval (0.9 seconds at 80 km/h in our experiments).

We argue that the supremum is an accurate representation of what is possible in a single pass because (1) the effects of the environment are relatively consistent across all runs, as discussed in Section 4.1.1, and (2) a reduction in goodput at a particular point on the roadway due to transient protocol behavior will be present in some runs but likely not all. Therefore the supremum goodput is at least a minimum for what is possible at each point along the roadway.

Theoretical Potential: Because the supremum can be limited by persistent problems that occurred in all runs, further utilization of the connection is possible. Therefore, we have also shown a rough computation of the theoretical potential in Figure 4.4, computed as follows. First, for each 20 m interval on the roadway, we were able to compute an estimate of the signal-to-noise ratio (SNR) based on the average measured signal energy (RSSI) and assuming a noise floor of -95 dBm (this is the noise floor assumed by the Madwifi driver). Next we computed the expected MAC bit rate at each point on the road based on the minimum receiver sensitivity for each MAC bit rate, as specified in the 802.11 standard [31]. Based on lab experiments we determined the TCP goodput possible for each MAC rate under ideal conditions, shown in Table 4.2. Summing the goodput over all intervals, we obtained a rough estimate of the theoretical potential data transferred in our environment, used here only as a point of reference.

As shown in Figure 4.4, existing networking protocols not only significantly underutilize connection potential by more than 50%, relative to the supremum, but also yield large variations in data transferred, despite all runs using identical configurations. Next we experimentally analyze the cause of this underutilization.

MAC Rate	TCP Goodput	MAC Rate	TCP Goodput
1 Mbps	0.75 Mbps	12 Mbps	8.8 Mbps
2 Mbps	1.53 Mbps	18 Mbps	12.7 Mbps
5.5 Mbps	3.88 Mbps	24 Mbps	16.4 Mbps
6 Mbps	4.51 Mbps	36 Mbps	22.4 Mbps
9 Mbps	6.76 Mbps	48 Mbps	27.7 Mbps
11 Mbps	6.87 Mbps	54 Mbps	29.3 Mbps

Table 4.2: Lab Measurements of Static TCP Goodput

4.3 Causes of Underutilization

We have identified ten problems that occur during three distinct phases of an opportunistic vehicular connection. Table 4.3 provides a roadmap for this section.

4.3.1 Overview

For the purposes of our analysis, we divide the connection into the entry phase, the production phase, and the exit phase, as done by Ott et al. [49], illustrated in Figure 4.5. During the entry phase, the connection is established and the signal slowly

Observed Effect	**Mechanism**	**Sec.**	**Fig.**	**Impact**
Entry Phase: Delayed Connection Startup (Section 4.3.2)	Lengthy AP Selection	4.3.2	4.9	Average Total Delay: 13.1 s ± 12.3 (95% CI) (Figures 4.7, 4.8), resulting in a median of 7.5% lost data transferred and 22.8% in 15% of runs
	MAC Management Timeout	4.3.2	4.10	
	Application Initialization Delay	4.3.2	4.8	
	ARP Timeout	4.3.2	4.11	
	Overestimation of Initial MAC Bit Rate	4.3.2	4.12	
	Early TCP Timeouts	4.3.2	4.11	
Production Phase: Underutilization of Performance Potential (Section 4.3.3)	TCP Sender: Slow Adaptation of MAC Bit Rate	4.3.3	4.2	Goodput would have been significantly reduced had we used the default MAC rate selection
	TCP Receiver: Slow Adaptation of MAC Bit Rate	4.3.3	4.15	16-23% goodput reduction
Exit Phase: Inefficient Use of Weakening Signal (Section 4.3.4)	Overestimation of MAC Bit Rate After the Production Phase	4.3.4	4.16	9.4 s ± 10.6 (95% CI) lost connection time, resulting in 2.1% median lost data transferred and 3.0% in 15% of runs
	TCP Timeout Near the Beginning of the Exit Phase	4.3.4	4.16, 4.17	

Table 4.3: Causes of Underutilization of Opportunistic Vehicular Connections

increases. Following that, the period of good connection quality, the production phase, is where the bulk of the data is transferred. After the production phase is the exist phase, where the connection quality decreases until the vehicle leaves the coverage range of the access point.

We have chosen to identify the production phase as beginning 320 meters before the access point and, for symmetry, ending 320 meters after the access point, as shown in Figure 4.5. This results in the three phases as follows:

1. Entry phase: starts at -640 meters, ends at -320 meters.
2. Production phase: starts at -320 meters, ends at 320 meters.
3. Exit phase: starts at 320 meters, ends at 640 meters.

During the production phase, the supremum goodput is greater than 3 Mbps. The choice of where to label the phase divisions is arbitrary and is only used for reference in our analysis. The total connection duration of the supremum is 58 seconds (at 80 km/h), with the three phases lasting 14.5 s (25%), 29 s (50%), and 14.5 s (25%),

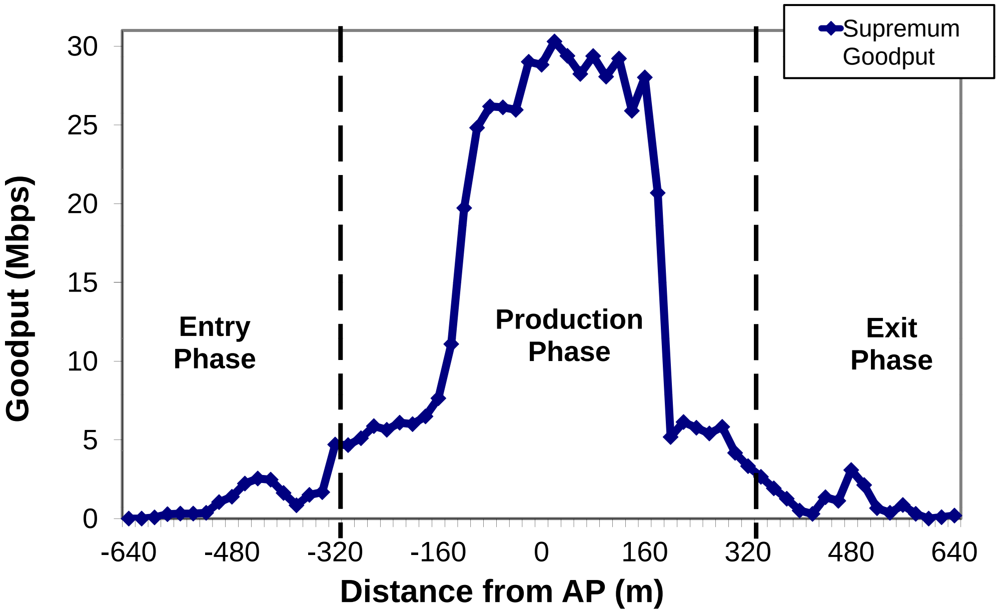


Figure 4.5: The three phases of a vehicular connection. The precise division between phases is arbitrary and is only used for reference in our analysis.

respectively. During the entry, production, and exit phases, 3%, 94%, and 3% of the supremum data transferred was transferred during each phase, respectively.

We next analyze the cause of connection underutilization in each of the three phases.

4.3.2 Entry Phase

As suggested in [49, 18, 10], high losses near the beginning of a connection could impair connection setup and significantly reduce the amount of data transferred during the connection. Here we quantify this through a detailed experimental analysis of the entry phase of an opportunistic vehicular connection.

Connection Setup Procedure

The connection setup procedure required to setup a TCP connection over an 802.11 link is shown in Figure 4.6. DHCP was not used in our experiments, as discussed in Section 3. As shown, there is a lengthy series of control messages that must be sent sequentially in order for a connection to be successfully set up. It is precisely this chattiness that causes problems in lossy environments, as identified by Zhuang et al. [76]. Recovering from the loss of an individual control message is handled by a variety of different mechanisms across different networking layers. In a lossy environment, like the entry phase, all of these mechanisms must function well together for quick connection setup; something not easily achieved using existing networking protocols, as shown next.

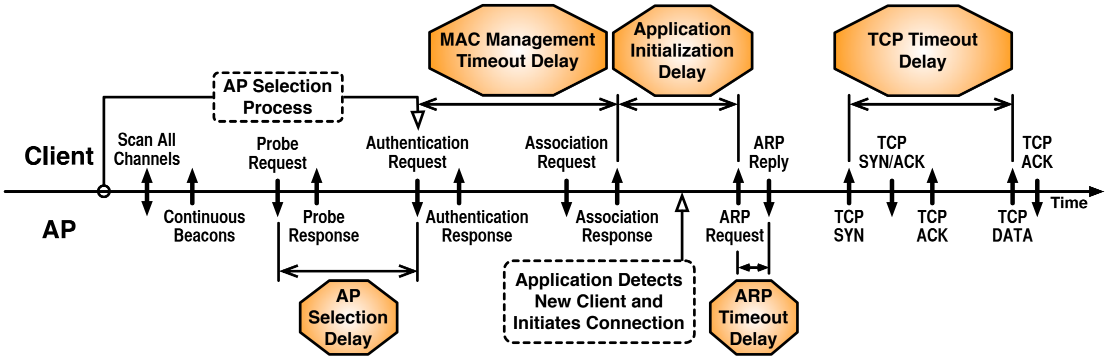


Figure 4.6: Timeline of connection setup procedure, showing the sources of delay.

Connection Setup Delay

Robust connection setup is crucial in order to fully utilize a short-lived opportunistic connection. Any delay in setting up the connection results in lost opportunity to send data, especially if it cuts into the production phase.

Figure 4.7 shows the locations where the connection setup completed. We consider connection setup to be complete once the first successful TCP Data and TCP ACK packets are exchanged. Figure 4.8 shows a breakdown of the major causes of connection setup delay. We examine these causes in detail next.

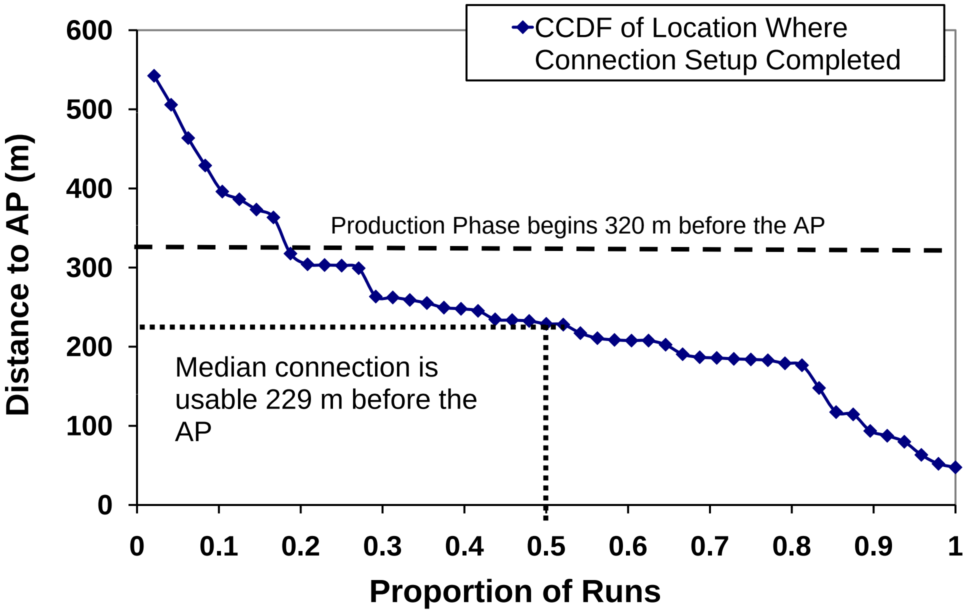


Figure 4.7: CCDF showing the location where the connection setup phase completed and the TCP connection became usable, indicating that less than 20% of connections made full use of the production phase.

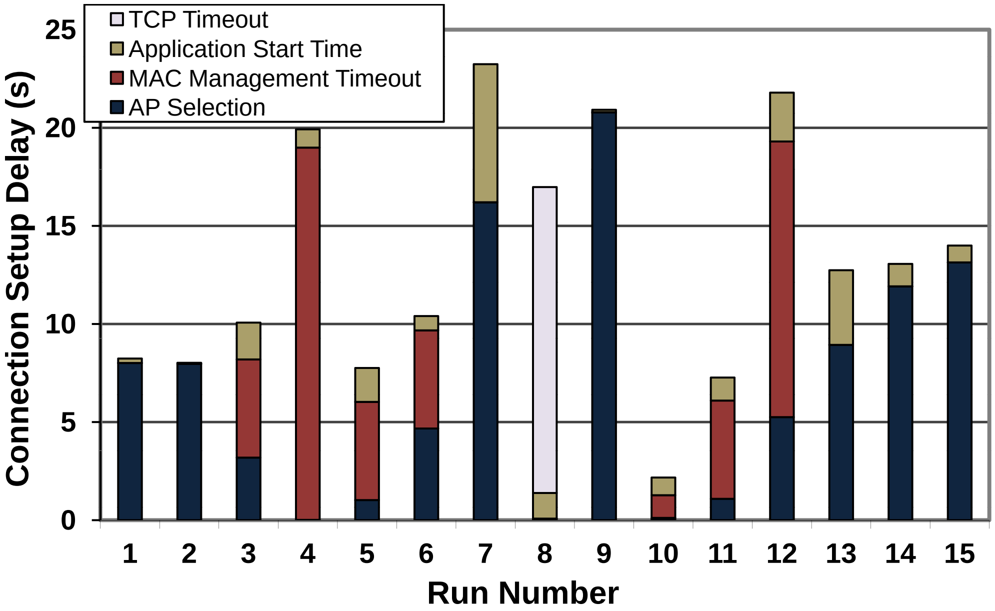


Figure 4.8: Cumulative bar graph showing the breakdown of the sources of delay for a random subset of the 48 experimental runs.

Lengthy Access Point Selection

Before an 802.11 MAC connection is attempted, a client must decide which access point to connection to. A client first locates all available access points by performing one or both of (1) a passive scan, involving sequentially switching through all channels and listening for beacon messages, (2) an active scan, where the client sends probe request messages and waits for probe responses from any available access point. Our cards performed both simultaneously.

The scanning process continues until the client locates an access point it wishes to connect to. Our experiments reveal that this process continues well after the roadside access point is first detected. In our experiments, the vehicular client received beacons from the access point very early on (750+ m before the access point), before a two-way connection was possible. This was due to the access point using a higher transmit power than the client (19 dBm vs 15 dBm, as discussed in Section 3).

As occurred in the run shown in Figure 4.9, the client continues to scan for alternative access points, even after a two way connection was possible. This is evident by the probe responses received, and acknowledged, by the client, which would often receive multiple probe responses from the roadside access point before deciding to associate.

The Madwifi driver decides to associate with an access point when the average measured signal energy (RSSI) over a fixed time interval crosses a certain threshold. This threshold, controlled by the `rssi11g` parameter, defaults to 24 RSSI units, which is the equivalent of -71 dBm of measured signal energy [35]. Because of the

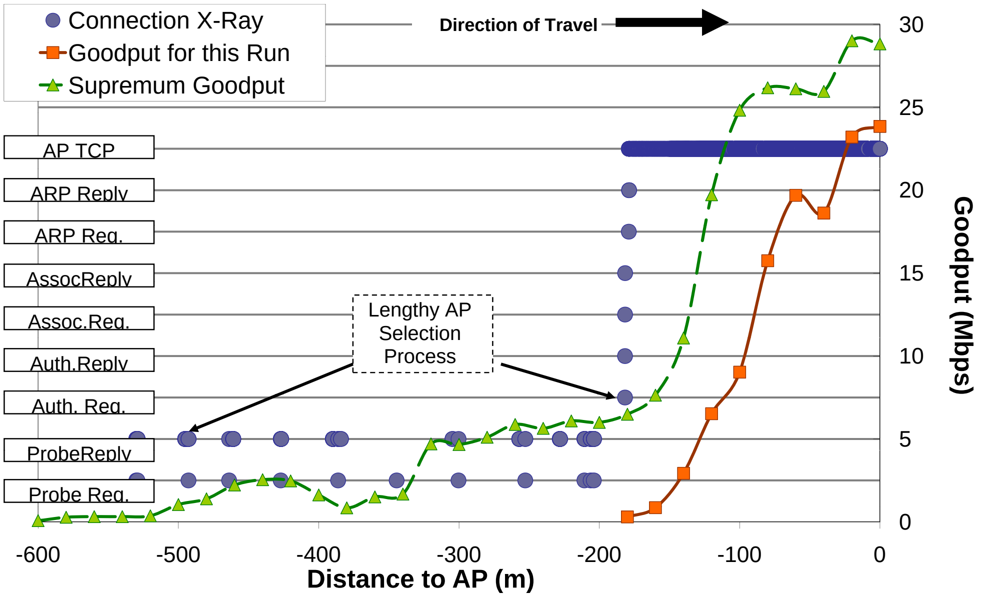


Figure 4.9: Example of Lengthy AP Selection

repeatability of the signal strength in this environment (recall Figure 4.1(a)), we expected that the AP selection process would be very consistent; however, Figure 4.8 shows that the length of the AP selection process was very inconsistent. We attribute this to rapid fluctuations of the instantaneous measured signal strength which caused the average signal strength to cross the association threshold (the `rssi11g` parameter) much earlier or much later in different runs.

MAC Management Timeout

As shown in Figure 4.6, once the client has decided to attempt association, it transmits a MAC authentication request. Because we are using open authentication and no encryption, this is followed by only three control messages to complete the MAC connection setup. However, if one of these messages is lost, the Madwifi driver recovers from this loss with a **hard-coded timeout of five seconds** and a single retransmission. After two successive losses, the AP selection process is restarted and this process is repeated. Figure 4.10 shows a drastic example of how losing a few MAC management messages results in a significant delay in connection setup. Although the five second timeout is specific to our hardware configuration, we discuss how this observation can be generalized in Section 5.1.

Application Initialization Delay

Because we did not use DHCP for IP address assignment, the next step in the connection setup procedure is for the access point to initiate a connection to the client's statically assigned IP address. Previous work has examined the delay due

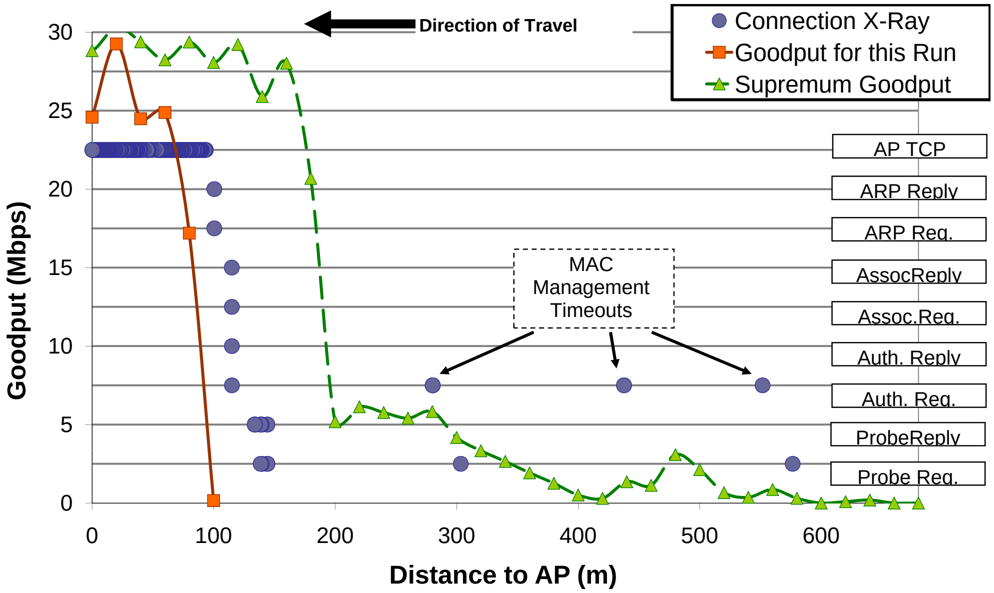


Figure 4.10: Example of MAC Management Timeout

to DHCP and found it to be 1.8 seconds on average [10]. This certainly would have a significant impact on connection setup; however, we chose to isolate other, lesser known, causes of delay in our work.

Once a MAC layer connection has been established, the application running on the access point must detect the new client and initiate a transfer to it. Previous work by Bychkovsky et al. [10] observed average application initialization delays of 5 seconds; however, they attributed this delay to the heavy load on their devices due to running database software. In our case, we launched iperf from a shell script loop, resulting in a small delay of 1-2 seconds, as shown in Figure 4.8.

ARP Timeout

Next, the access point must perform an ARP lookup of the client's MAC address based on the destination IP address requested by the application. Figure 4.11 shows that lost ARP messages are retransmitted after one second, a reasonable timeout for a vehicular scenario. Therefore, we found that ARP timeouts were not a significant source of connection setup delay.

Overestimation of Initial MAC Bit Rate

When a MAC layer connection first begins, an initial MAC bit rate must be chosen before any feedback from the environment is obtained. The SAMPLE rate selection algorithm we used chose an initial rate of either 11, 36, or 54 Mbps using simple IF-statements based on signal strength. Analysis of our logs revealed that it was

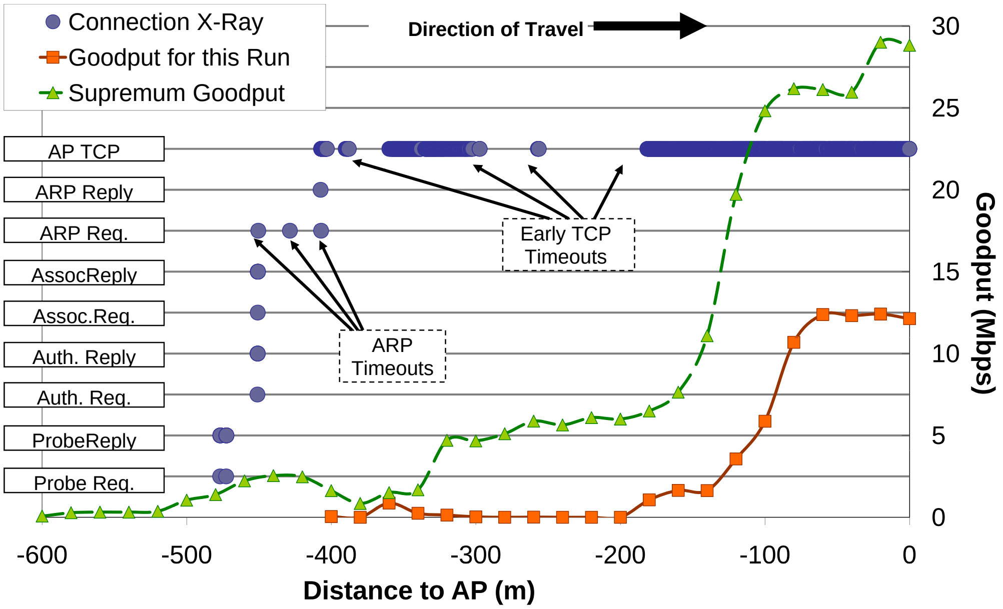


Figure 4.11: Example of two ARP timeouts, causing a two second delay. Early TCP timeouts also occurred during this experimental run, causing a 10.5 second delay, discussed later.

not uncommon for high initial rates to be selected, even though the connection was clearly not capable of transmitting at those high rates. The decision to use a high initial bit rate is not unique to the SAMPLE algorithm. As shown in Table 4.4, many other algorithms do the same.

Algorithm	**Initial Bit Rate**
SAMPLE [7]	11, 36, or 54 Mbps depending on signal strength
Onoe [44]	36 Mbps
AMRR [39]	36 Mbps
RRAA [71]	54 Mbps

Table 4.4: Initial MAC Bit Rate Used by Rate Selection Algorithms

Choosing an initial bit rate that is higher than the wireless channel can support introduces a delay before the connection becomes useable, as the bit rate algorithm must gradually reduce its rate until it matches the channel's capability. In our experiments using the improved SAMPLE parameters, this still took up to six seconds, as was the case in the run shown in Figure 4.12.

Although overestimating the initial bit rate causes high MAC losses, we found that this did not necessarily translate into high TCP losses. We attribute this to the Atheros multi-rate retransmission mechanism that reduces the bit rate for successive retransmissions, as discussed in Section 4.1.3. However, because high rates are used when lower rates would have been more suitable, the amount of

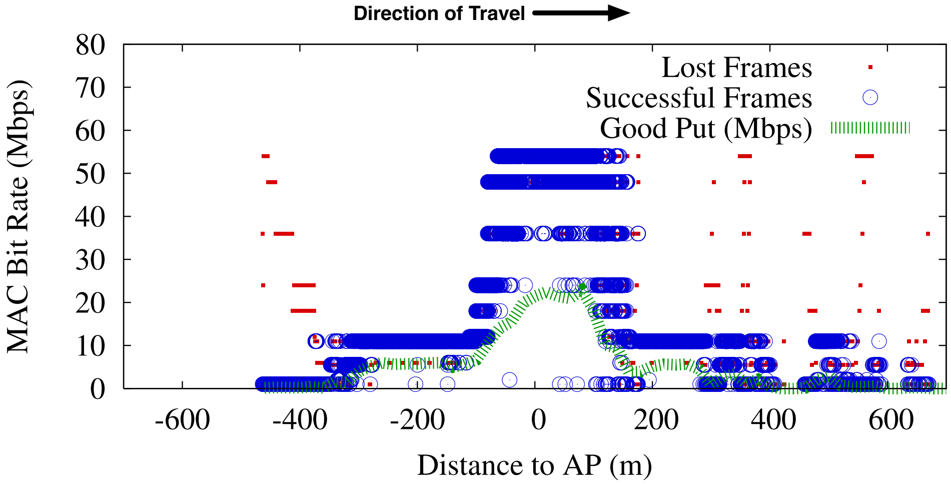


Figure 4.12: Example of overestimation of initial MAC bit rate; 54 Mbps was chosen although only 1 Mbps was possible during the initial phase of the connection. In this run it took 115 m (6 s) to adapt to the proper bit rate. Also seen later in the connection are the frames used to probe higher rates.

viable transmission attempts is reduced and therefore high MAC losses due to overestimation of bit rate does increase the probability of TCP losses.

Early TCP Timeouts

As discussed previously in Section 4.1.3, the overall number of losses seen by TCP was very low because of the aggressive 802.11 MAC retransmission scheme. However, the losses that are seen by TCP are more likely to occur during periods of weak signal such as during the entry or exit phase as there is a higher probability that all of the MAC retransmissions will fail.

During the early phase of an opportunistic connection, TCP is more likely to enter a back-off state because (a) its congestion window is small and therefore TCP's fast retransmit mechanism is less likely to be engaged, and because (b) losses are higher at the fringe of coverage. Entering a back-off state is effectively equivalent to pausing transmission at the TCP sender, something highly undesirable during a short-lived connection.

Although we did observe early TCP timeouts, as seen in the previously shown run in Figure 4.11, they were not as common as we had anticipated. The previously shown Figure 4.8 shows that an early TCP timeout only occurred in one of the 15 runs shown. We attribute this to (a) better than expected connection quality once the TCP connection was established, due to the TCP connection starting relatively late because of the MAC layer delays discussed previously and (b) the aggressive Atheros MAC retransmission scheme, discussed in Section 4.1.3. If the MAC layer delays were reduced, we would expect to see an increased amount of TCP timeouts during the early phase of the connection.

Understanding the Effects of Connection Setup Delay

Although it appears from our results (particularly Figure 4.8) that some sources of delay are more significant than others, we argue that they are highly dependent on one another. That is, had one source of delay not been present, the one following it would likely have taken its place. For example, had the AP selection process not lasted as long, more MAC management timeouts would likely have occurred due to weak signal, consuming approximately the same delay as before. The same argument can be extended to ARP timeouts and TCP timeouts. As a result, a solution to reduce connection setup delays must be all-or-nothing, as the connection cannot become useful unless all delays are addressed. We discuss this and other possible solutions in Section 4.4.

Impact of Connection Setup Delay

As shown in Figure 4.7, connection setup delay prevents most connections from becoming usable until well into the production phase. In order to gauge the impact of this, we determined the lost potential data transferred that resulted from late connection setup. Using the supremum goodput (Figure 4.1(c)), we found that for the median connection setup distance, 7.5% of potential data transferred was lost. In 15% of runs, more than 22.8% was lost, and in the worst case 40.3% was lost.

4.3.3 Production Phase

TCP Sender: Slow Adaptation of MAC Bit Rate

The first major problem we encountered was that the default parameters used by the default bit rate selection algorithm used in our equipment did not perform well in the vehicular environment (discussed previously in Section 4.1.2), requiring us to modify the algorithm's parameters to make it more dynamic for our environment.

Although the results in this thesis were gathered using our modified parameters of the bit rate algorithm on the access point, early experiments revealed that using the default parameters resulted in up to four times less data transferred than using our modified parameters. Thus, using a bit rate selection algorithm on the sender that is well suited to the environment is critical.

Next we report on the impact using the default algorithm parameters on the unmodified clients.

TCP Receiver: Slow Adaptation of MAC Bit Rate

Recall from Section 3.2 that our goal in these experiments was to analyze performance characteristics experimentally using unmodified clients. This represents the limit of what an infrastructure provider has control over. As a result, we chose to use the default bit rate selection parameters on the clients rather than our modified parameters.

As expected, the TCP receiver (the vehicle) tended to use lower bit rates for sending TCP ACKs (Figure 4.14) than the access point for sending TCP Data (Figure 4.13). Here we notice two unusual spikes in the rates used by the client. This is a result of the slow adaptation of the default rate selection algorithm, which bases its decisions on a 10 second history (recall Table 4.1). As is the case in the run shown in Figure 4.15, the rate was not increased until after it had passed the access point.

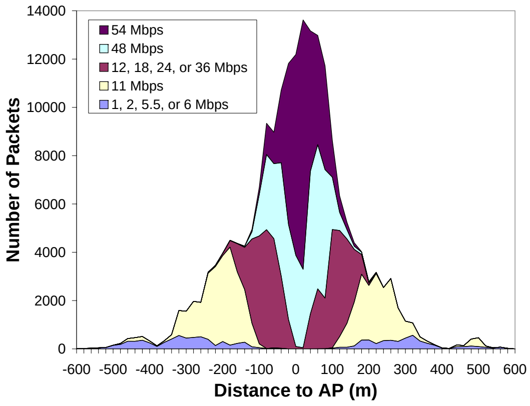


Figure 4.13: MAC bit rates used by the access point (TCP sender) at each section of the roadway, summed across 15 vehicular runs, demonstrating the repeatability of the connection

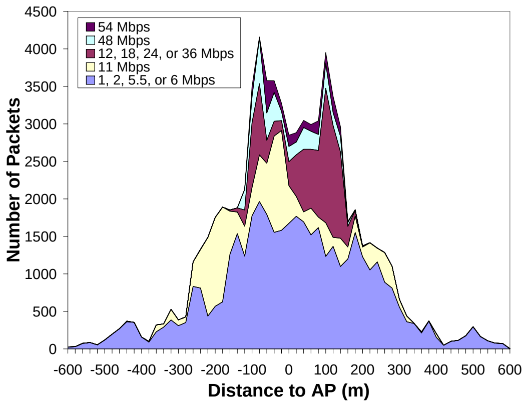


Figure 4.14: MAC bit rates used by the vehicular client at each section of the roadway for sending TCP ACKs, summed across 15 runs. The unusual two peaks in the curve are explained in Section 4.3.3.

However, this did not occur during every run, as evident by the large error bars in Figure 4.1(c). A closer examination of our logs revealed that in some runs, the client used very high rates (e.g., 54 Mbps) for TCP ACKs, and in others, much lower rates were used (e.g., 11 Mbps). Lower rates (such as in the run shown in Figure 4.15) were common due to the slow adaptation of the default bit rate selection parameters. However, in other runs, higher rates were achieved because of a combination of two factors: delayed connection setup and a high initial bit rate. If connection setup was sufficiently delayed such that the MAC connection did not complete until after the entry phase, then the initially high bit rate used by the client would succeed rather than being immediately reduced by the rate selection algorithm.

We discovered that using the default bit rate algorithm on the client had a much larger impact than expected. The slower rate used for TCP ACKs consumed more air time and resulted in reduced goodput. In order to quantify the impact of this, we first determine the overall degradation in the production phase by identifying the

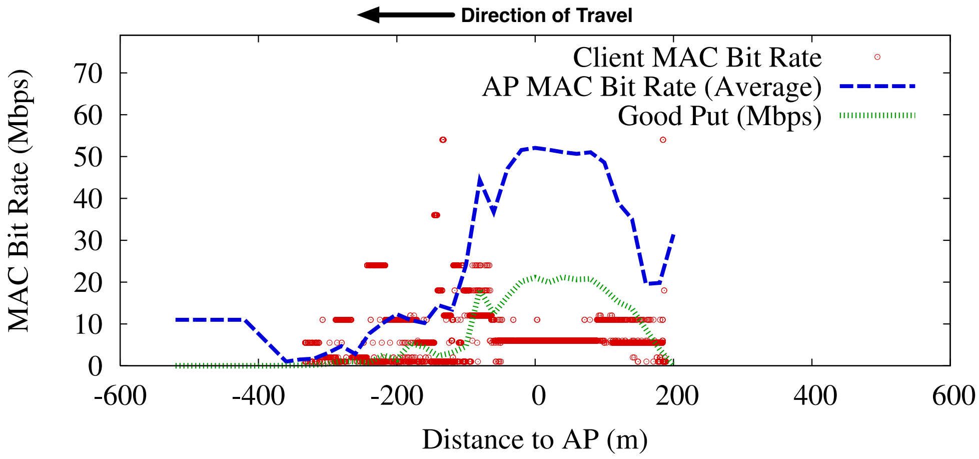


Figure 4.15: Example run showing slow adaptation of MAC bit rate on vehicular client (TCP receiver); the average bit rate used by the AP for this run is shown for reference

difference between the average goodput and supremum goodput during the middle of the production phase. The average goodput is 25% less than the supremum, as can be verified visually in Figure 4.1(c).

Next we examine the bit rates used, by the access point and the client, aggregated across all runs, shown in Figures 4.13 and 4.14 respectively. The first immediate difference is that, during the middle of the production phase, rates 54 and 48 Mbps were used the majority of the time by the access point and 2, 6, and 11 Mbps were used the majority of time by the client.

Lab experiments, shown in Table 4.5, reveal that the air time used by the TCP ACKs has a significant impact on overall TCP goodput. Based on these measurements and the proportion of rates used by the access point and the client, we compute that the lower bit rates used on the client account for approximately 19% of lost goodput during the middle of the production phase.

Sender MAC Rate	**Receiver MAC Rate**	**TCP Goodput**	**% of Max**
54 Mbps	1 Mbps	11.3 Mbps	39%
54 Mbps	2 Mbps	16.5 Mbps	56%
54 Mbps	5.5 Mbps	20.5 Mbps	70%
54 Mbps	11 Mbps	22.3 Mbps	76%
54 Mbps	6 Mbps	25.2 Mbps	86%
54 Mbps	9 Mbps	26.5 Mbps	90%
54 Mbps	12 Mbps	27.5 Mbps	94%
54 Mbps	54 Mbps	29.3 Mbps	100%

Table 4.5: Effect of MAC bit rate used for TCP ACKs (lab measurements). The first four receiver MAC rates shown consume more air time than the remainder due to using the legacy 802.11b preamble and timing.

We argue that this can be reasonably extrapolated to the remainder of the production phase, and because 94% of the data was transferred during the production phase, we conclude that using the default bit rate selection on the TCP client (the vehicle) resulted in approximately 16% to 23% less data transferred compared to the supremum.

4.3.4 Exit Phase

Overestimation of MAC Bit Rate After the Production Phase

After the production phase, connection quality begins to decrease. We found that the MAC bit rate selection algorithm on the access point, even with our improved SAMPLE parameters, failed to adequately adjust to the decreasing signal quality. This resulted in the same overestimation symptoms experienced during the entry phase, as discussed in Section 4.3.2. Figure 4.16 shows an example of the high bit rates attempted by the access point after the production phase. A more dynamic bit rate selection algorithm is needed to adjust to the rapidly changing channel conditions in this environment.

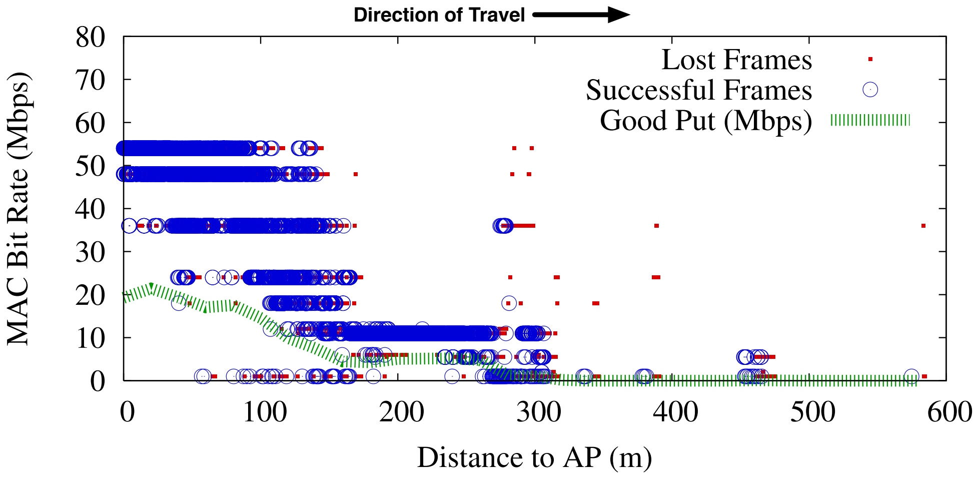


Figure 4.16: Example run showing overestimation of MAC bit rate after production phase causing a TCP timeout at 300 m (last half of connection shown).

TCP Timeout Near the Beginning of the Exit Phase

Much like the entry phase, due to overestimation of MAC bit rate and poor connection quality, TCP losses were likely to occur in the exit phase. The earlier the TCP timeout occurred, the more potential data transferred was lost. Figure 4.16 shows an example of a TCP timeout that occurred shortly after the production phase, 300 m past the access point.

Figure 4.17 shows the CCDF of the locations where the last useful TCP packet was received by the vehicle. Here we see that the majority of the exit phase was unused in most runs, as the supremum exit phase ended at 640 m past the access point while the median connection ended much earlier, at 398 m, representing an average of 9.4 seconds ± 10.6 (95% CI) of lost connection time. Compared to the supremum, this resulted in a median loss of 2.1% potential data transferred, 3.0% in 15% of runs, and 9.2% in the worst case.

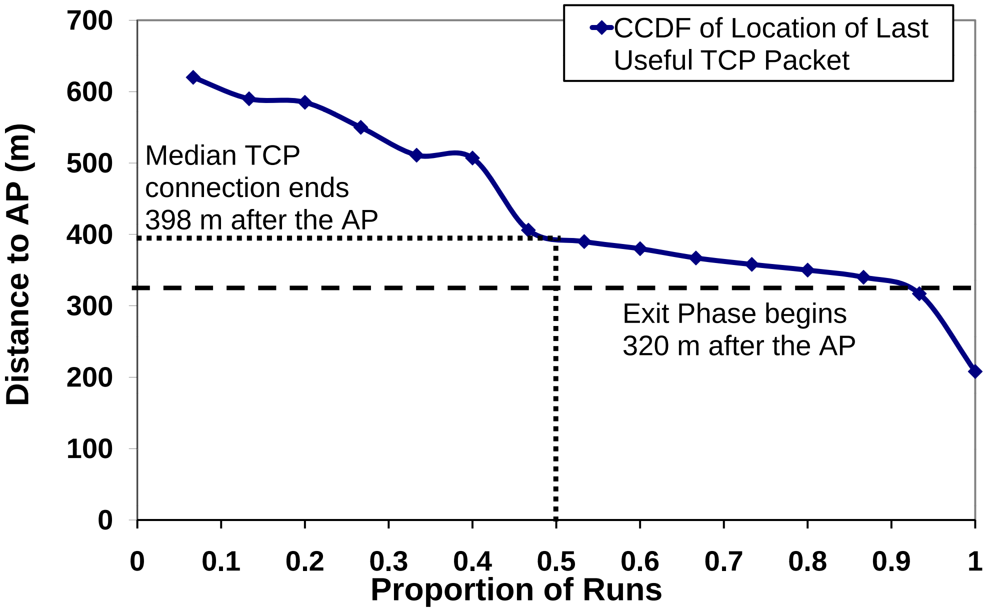


Figure 4.17: CCDF showing the location where the TCP connection became unusable. As shown, the majority of runs ended early into the exit phase.

4.4 Recommendations

Based on our findings, we now suggest some best practices to other system implementers for improving vehicular opportunistic communication using existing 802.11 hardware. Our recommendation is make full use of the production phase by ensuring rapid connection setup and using a dynamic bit rate selection algorithm.

As shown in Section 4.3 the two major causes for poor use of the production phase are: (1) connection setup delays, lasting well into the production phase, effectively reducing the length of the production phase; and (2) the default client bit rate selection algorithm leads to sub-optimal use of the production phase. Therefore, our recommendations translate to mechanisms to mitigate these problems.

4.4.1 Reducing Delays in Connection Setup

To prevent the connection setup delays from cutting into the production phase, the simplest solution is to have mobile devices **avoid the fringe area**. That is, a device should not attempt to use an AP until the start of the production phase.

Of course, this raises the question of how can a device know that the production phase has started. There are several possible approaches. For instance, the device could attempt to associate with an AP only when the RSSI exceeds some threshold. Indeed, our mobile device's access point selection process already uses such a threshold. Unfortunately, a single packet with an anomalously high RSSI value is enough to kick off the association process. Therefore, the threshold has to be combined with some degree of signal filtering. Another approach would be to periodically probe for an AP using an 802.11 probe packet, but with a very short probe timeout. While effective, this expends more power, which may preclude its use in some situations. A well-considered solution for reliably detecting the production phase appears to be a fruitful area for future work.

4.4.2 MAC Bit Rate Selection Algorithms

We saw in Section 4.3.3 that using the default bit rate selection algorithm reduces goodput in the production phase. Because the bit rate selection algorithms used by cards are generally kept secret by chip makers, and could potentially be different between chip revisions, the only choice a system builder has is to purchase multiple cards and use whichever card works best. We recommend that system designers use this pragmatic step when building opportunistic communication systems.

Should the system designer have the ability to tune the bit rate selection algorithm, we suggest making the algorithm more dynamic, as was done for our experiments (explained in Section 4.1.2).

4.4.3 Tuning Parameters

If system builders cannot stop a card from using the fringe, but have the ability to change parameters on a device, they can still fine-tune their data-link and transport layer protocols to work better in the vehicular environment. Some useful tweaks that could be made are: (a) increasing the minimum RSSI needed before connecting to an access point, (b) reducing the MAC management timeout, (c) reducing the ARP timeout, (d) using a more dynamic MAC bit rate selection algorithm, and (e) reducing TCP's initial timeout value, and (f) making TCP retransmissions more aggressive.

Note that there is a limit to the effectiveness of tuning the last hop of the connection (either the access point or the wireless client) because the TCP sender could be any host on the Internet, which cannot be controlled. Therefore, we advocate using a connection-splitting approach, such as done by Indirect-TCP [1], to split the TCP connection into two halves at the access point, allowing the access point to control the TCP parameters used over the wireless link.

To sum up, we recommend that system implementers either avoid the fringe, or, if that is not feasible, reduce the effect of the fringe on protocol performance by fine-tuning parameters. Performance can also be improved by evaluating multiple cards and multiple firmware versions.

4.5 Discussion: the Role of Environmental Awareness

Although the focus of this thesis is on understanding the problems associated with vehicular opportunistic connections, we believe that our work raises the broader question of the role of environmental awareness in networking protocols. To see this, consider first that all wireless technology deployments, including those of the future, are likely to contain areas with marginal coverage, where packet losses are very high due to dead spots, weak spots, and interference. Section 4.3.2 showed that existing data-link and transport layer protocols perform poorly in marginal coverage areas. Worse, today's networking protocols are not only poor at dealing with bursty losses, but also hide their failures from the layer above, causing destructive protocol interactions.

To avoid these problems, future data-link and transport layer protocols should be less sensitive to bursty wireless losses, especially during control-plane actions, such as association and authentication. They should also use initial operating parameters that are suitable for marginal coverage areas. Figure 4.18 illustrates the effects of not selecting appropriate initial operating parameters. If it is not possible to adapt parameters, communication protocols should try to use the wireless channel for control actions only when it is known to be in a good state. In any case, they should report a failure to an upper-layer protocol immediately, so that it can take the appropriate action.

At a more abstract level, we believe that future networking protocols need to be more aware of their operating environment. This would allow them to (1) better choose initial operating parameters; and (2) better deal with very high variability in packet loss rates (or, equivalently, packet delays), and link capacities. Environmental awareness could be accomplished either manually, by a user preference setting, or automatically by some learning or detection process on the client or the access point.

In this light, it is interesting to note the *privileged position of a roadside AP* in vehicular opportunistic communication. It participates in every communication and can therefore exploit its knowledge of past connection history to help future connections. This is particularly powerful because signal strength is relatively consistent

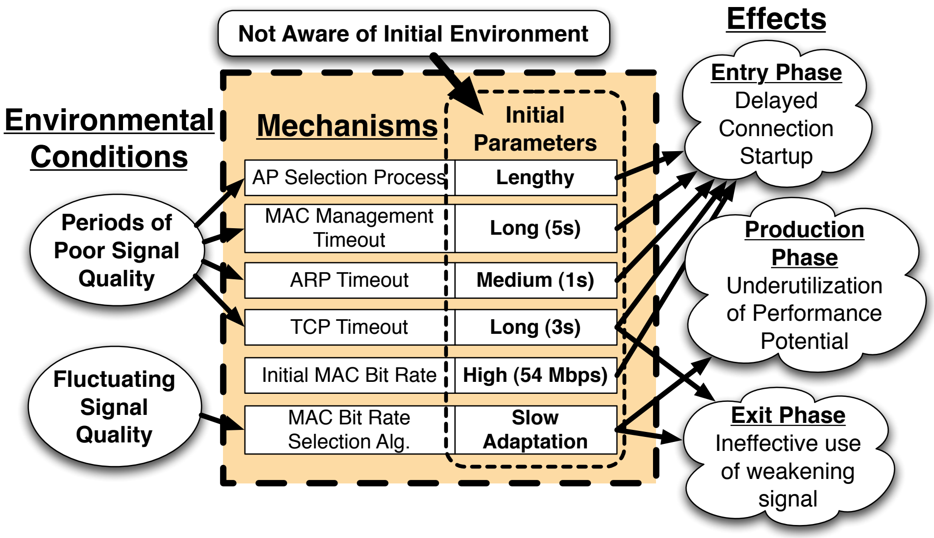


Figure 4.18: The cause-mechanism-effect relationship resulting in underutilization of opportunistic vehicular connections.

between vehicular passes (as discussed in Section 4.1.1). For example, suppose the access point recorded the average signal strength of a client relative to its GPS position. The access point could then (1) build an approximate picture of the pattern of signal quality and (2) determine how rapidly the signal quality changes, perhaps due to client mobility. The access point could then use this information to adjust its operating parameters, such as setting the initial bit rate, using an appropriately aggressive bit rate selection algorithm, and adjusting MAC and TCP timeouts to make them more suitable for the environment. Moreover, it could even give hints about operating parameters to incoming vehicles as a field in its beacon messages.

In summary, we believe that heightened environmental awareness will be a key feature of future communications protocols and that, in the context of roadside communication, the AP can play a critical role in bringing this about.

Chapter 5

Conclusions

The feasibility of opportunistically using 802.11 hotspots for vehicular Internet access has been confirmed by previous work [49, 27, 18, 10, 42]. In this thesis, we build on this work with a detailed experimental analysis of data-link and transport layer protocol behavior during opportunistic vehicular communication at a depth not previously examined. We show that heightened awareness of the operating environment, particularly in the vehicular scenario, can dramatically increase the overall throughput of a connection.

In our experiments, we show that the amount of data transferred during an opportunistic connection was only 50% of what was possible in this scenario. Specifically, we show that losses during the connection setup phase contribute significantly to this amount.

We have identified ten problems that cause this throughput reduction during the three phases of an opportunistic connection, including:

1. *Entry phase:* Problems in the entry phase caused a median loss of 7.5% of potential data transferred, 22.8% was lost in 15% of runs, and 40% was lost in the worst case, due to:

    (a) Lengthy AP selection process

    (b) Long MAC management timeouts

    (c) Application initialization delay

    (d) ARP timeouts

    (e) Overestimation of the initial MAC bit rate

    (f) TCP timeouts early in the entry phase

2. *Production phase:* 16% to 23% less data was transferred due to problems during the production phase.

    (a) Significantly less data would have been transferred had we used the default parameters of the default rate selection algorithm (SAMPLE) on the access point.
    (b) 16% to 23% less data was transferred than possible due to slow adaptation of MAC bit rate on the default client during the production phase.

3. *Exit phase:* Problems in the exit phase caused a median loss of 2.1% potential data transferred, 3.0% was lost in 15% of runs, and 8.4% was lost in the worst case, due to:

    (a) Overestimation of MAC bit rate after the production phase
    (b) TCP timeouts near the beginning of the exit phase

We further suggest best practices for vehicular opportunistic connections. First, we argue that the best way to use current networking protocols in this scenario is to avoid the fringe areas altogether, due to protocol timeouts and back-off procedures extending into the production phase. Second, we recommend a fruitful direction of future protocol design which involves exploiting environmental knowledge to optimize protocol behavior for the operating environment. In the case of opportunistic vehicular data transfers, our experimental analysis shows that overall throughput could be improved by up to a factor of two by using such environmental information.

5.1 Limitations and Future Work

The biggest limitation of our work in this thesis is that we have only evaluated a single scenario: one vehicle, one vehicle speed, one environment, one wireless card, and only downlink data transfers that used only TCP. However, some of these aspects of opportunistic connection have already been examined. Previous work [49, 27, 18] has shown that data transferred is inversely proportional to vehicle speed. As well, in [27], we examined data transfer in the uplink direction and found similar behavior to that in the downlink direction. Furthermore, in ongoing work, we have experimented with UDP transfers and preliminary results show very little difference between UDP and TCP due to the aggressive retransmissions done at the MAC layer. Bychkovsky et al. [10] and Mahajan et al. [42] have studied vehicular opportunistic connections in an urban setting; however, further exploration is needed to understand the details of data-link and transport layer behavior in an urban environment.

Gaining a thorough understanding of the behavior of different wireless cards in this scenario is a fruitful direction of future work. Existing work [21] indicates that there are subtle differences in the implementation of the 802.11 standard between chip manufacturers that can have a significant impact on how they communicate. Because of the short duration of a vehicular connection, the effects of these subtleties is likely to be exaggerated.

A large space of future work lies in designing an 802.11 bit rate selection algorithm for the vehicular environment. Although mobility-aware 802.11 rate selection algorithms have been developed in recent literature [71, 55, 38], none have adequately addressed the needs of high-speed vehicular communication. Specifically, because of the short connection duration and rapid fluctuations in signal quality, probing different rates, as is commonly done, may lead to significant inefficiencies in this scenario.

Understanding protocol behavior in the presence of multiple vehicles is a fruitful avenue of future work. The effects of the 802.11 performance anomaly [28] on data transferred in scenarios with multiple vehicles has been examined by Hadaller et al. [25] using trace-based simulations. However, much work remains on understanding detailed protocol interaction and potentially designing new communication protocols that are better suited to multi-vehicle communication. In particular, when a second vehicle enters coverage, the communication of an earlier vehicle near the access point will effectively drown-out the second vehicle's attempts to associate with the access point, preventing the second vehicle from being able to fairly use the connection opportunity.

Although we have identified potential gains from heightened environmental awareness, implementing such ideas is non-trivial. One question that needs to be answered is what is a good source of environmental information? Should it be automatically detected or should the user manually indicate the operating conditions (e.g., stationary, mobile, highly mobile)? A second question is what is the best way to get this information to the networking protocols? Should the protocol API be changed to allow the input of these parameters, or should the operating system provide a common interface for applications to tune networking parameters? These are avenues of future work.

A higher layer problem, which has been a particular challenge for delay tolerant networking research [15, 58, 59], is how to use intermittent connectivity at the application layer. Existing applications assume a persistent and reliable underlying connection and cannot handle frequent disruptions. Making applications disconnection-aware is a broad area of future work.

References

[1] A. Bakre and B. R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 1995. 14, 51

[2] H. Balakrishnan and R. Katz. Explicit Loss Notification and Wireless Web Performance. In *IEEE Global Communications Conference (GLOBECOM)*, 1998. 14

[3] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997. 14

[4] H. Balakrishnan, S. Seshan, and R. Katz. Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks. *ACM Wireless Networks*, 1(4):469–481, 1995. 7, 14

[5] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. Levine, and J. Zahorjan. ViFi: Interactive WiFi Connectivity for Moving Vehicles. In *ACM Special Interest Group on Data Communications Conference (SIGCOMM)*, 2008. 12

[6] N. Banerjee, M. Corner, and B. Levine. An Energy-Efficient Architecture for DTN Throwboxes. In *IEEE Conference on Computer Communications (Infocom)*, 2007. 9

[7] J. Bicket. Bit-rate Selection in Wireless Networks. Master's thesis, MIT, 2005. 15, 31, 43

[8] BitTorrent. http://www.bittorrent.com/. 12

[9] J. Borras and R. Yates. Infostation Overlays in Cellular Systems. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 1999. 10

[10] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2006. 2, 11, 19, 38, 42, 53, 54

[11] R. Chakravorty, J. Cartwright, and I. Pratt. Practical experience with TCP over GPRS. In *IEEE Global Communications Conference (GLOBECOM)*, 2002. 2

[12] C. Chen. A Routing Framework for Interplanetary Internet. Technical report, Georgia Institute of Technology, 2005. 13

[13] D. DePriest. NMEA Data. http://www.gpsinformation.org/dale/nmea.htm. 22, 24

[14] Dedicated Short Range Communication (DSRC) group. http://grouper.ieee.-org/groups/scc32/dsrc/index.html. 9

[15] DTN Research Group (DTNRG). http://www.dtnrg.org/. 8, 55

[16] Fleetnet. http://www.fleetnet.de/. 10

[17] R. Frenkiel, B. Badrinath, J. Borres, and R. Yates. The Infostations Challenge: Balancing Cost and Ubiquity in Delivering Wireless Data. *IEEE Personal Communications*, 7:66–71, 2000. 1, 9

[18] R. Gass, J. Scott, and C. Diot. Measurements of In-Motion 802.11 Networking. In *IEEE Workshop on Mobile Computing System and Applications (HOTMOBILE)*, 2006. 2, 10, 11, 38, 53, 54

[19] Globalsat Technology Corporation. http://www.globalsat.com.tw/. 21

[20] T. Goff, J. Moronski, D. Phatak, and V. Gupta. Freeze-TCP: A True End-to-end TCP Enhancement Mechanism for Mobile Environments. In *IEEE Conference on Computer Communications (Infocom)*, 2000. 7

[21] K. Gopinath, P. Bhagwat, and K. Gopinath. An Empirical Analysis of Heterogeneity in IEEE 802.11 MAC Protocol Implementations and its Implications. In *MobiCom Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, 2006. 15, 54

[22] GPS SPS Signal Specification, 2nd Edition (June 2, 1995). http://www.-navcen.uscg.gov/pubs/gps/sigspec/default.htm. 21

[23] M. Guo, M. Ammar, and E. Zegura. V3: A Vehicle-to-Vehicle Live Video Streaming Architecture. In *Conference on Pervasive Computing and Communications (PerCom)*, 2005. 9

[24] D. Hadaller. Determining Precise Distance in a Highly Mobile Environment using Imprecise GPS Measurements. In *ACM MobiSys Workshop on System Evaluation for Mobile Platforms (MobiEval)*, Invited Talk: San Juan, Puerto Rico, 2007. 3

[25] D. Hadaller, S. Keshav, and T. Brecht. MV-MAX: Improving Wireless Infrastructure Access for Multi-Vehicular Communication. In *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, 2006. 15, 55

[26] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular Opportunistic Communication Under the Microscope. In *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2007. 3, 15

[27] D. Hadaller, H. Li, and A. Sung. Drive By Downloads: Studying Characteristics of Opportunistic Connections. In *USENIX Networked Systems Design and Implementation (NSDI) Poster Session*, 2005. 2, 11, 53, 54

[28] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *IEEE Conference on Computer Communications (Infocom)*, 2003. 15, 55

[29] Host Identity Protocol (RFC 5201). 7

[30] A. Iacono and C. Rose. Bounds on File Delivery Delay in an Infostations System. In *IEEE Vehicular Technology Conference (VTC)*, Spring 2000. 10

[31] IEEE Standard 802.11a Specification (8802-11:1999/Amd 1:2000(E)). http://-ieeexplore.ieee.org/servlet/opac?punumber=7118. 2, 36

[32] Mobile IP. RFC 3344. 7

[33] Iperf. http://dast.nlanr.net/Projects/Iperf/. 20

[34] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *ACM Special Interest Group on Data Communications Conference (SIGCOMM)*, 2004. 13

[35] Joe Bardwell. Converting Signal Strength Percentage to dBm Values. http://-www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength-.pdf. 40

[36] E. Jones, L. Li, and P. A. S. Ward. Practical Routing in Delay-Tolerant Networks. In *SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005. 13

[37] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Shiuan Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, 2002. 12

[38] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2008. 55

[39] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, 2004. 15, 43

[40] Y. Lee. Measured TCP Performance in CDMA 1x EV-DO Network. In *Passive and Active Measurement Conference (PAM)*, 2006. 2

[41] J. Leva, M. de Haag, and K. Van Dyke. Performance of Standalone GPS. *In: Understanding GPS Principles and Applications, edt. E. Kaplan and C. Hegarty, Artech House Inc., Norwood, USA*, pages 237–320, 1996. 21

[42] R. Mahajan, J. Zahorjan, and B. Zill. Understanding WiFi-based Connectivity From Moving Vehicles. In *ACM Internet Measurement Conference (IMC)*, 2007. 2, 11, 53, 54

[43] L. Michael and M. Nakagawa. Multi-hopping Data Considerations for Inter-vehicle Communication over Multiple Lanes. In *IEEE Vehicular Technology Conference (VTC)*, 1997. 9

[44] Multiband Atheros Driver for WIFI. http://www.madwifi.org/. 15, 18, 34, 43

[45] A. Munaretto, M. Fonseca, K.A. Agha, and G. Pujolle. Fair Time Sharing Protocol: A Solution for IEEE 802.11b Hot Spots. *Springer Lecture Notes in Computer Science*, 3124:1261 – 1266, 2004. 15

[46] A. Nandan, S. Das, G. Pau, M. Gerla, and M.Y. Sanadidi. Co-operative Downloading in Vehicular Ad-hoc Wireless Networks. In *IEEE Wireless On demand Network Systems and Services (WONS)*, 2005. 12

[47] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das. Mobisteer: Using Directional Antenna Beam Steering to Improve Performance of Vehicular Internet Access. In *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2007. 12

[48] Network Mobility (NEMO) Basic Support Protocol (RFC 3963). 7

[49] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for Automobile Users. In *IEEE Conference on Computer Communications (Infocom)*, 2004. 2, 10, 11, 36, 38, 53, 54

[50] J. Ott and D. Kutscher. A Disconnection-Tolerant Transport for Drive-thru Internet Environments. In *IEEE Conference on Computer Communications (Infocom)*, 2005. 10

[51] J. Ott and D. Kutscher. A Mobile Access Gateway for Managing Intermittent Connectivity. In *IST Mobile and Wireless Communication Summit*, 2005. 10

[52] J. Ott and D. Kutscher. Towards Automated Authentication for Mobile Users in WLAN Hot-Spots. In *IEEE Vehicular Technology Conference (VTC)*, Fall 2005. 10

[53] J. Ott and D. Kutscher. Bundling the Web: HTTP over DTN. In *ICST Workshop on Networking in Public Transport (WNEPT)*, 2006. 10

[54] J. Ott and D. Kutscher. Service Maps for Heterogeneous Network Environments. In *IEEE Mobile Data Management Conference (MDM)*, 2006. 10

[55] K. Ramachandran, R. Kokku, H. Zhang, and M. Gruteser. Synchronous Two-phase Rate and Power Control in 802.11 WLANs. In *ACM Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2008. 55

[56] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 2nd edition, 2001. 23

[57] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based Characterization of 802.11 in a Hotspot Setting. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, 2005. 19

[58] J. Scott, P. Hui, J. Crowcroft, and C. Diot. Haggle: A Networking Architecture Designed Around Mobile Users. In *IFIP Conference on Wireless On demand Network Systems (WONS 2006)*, 2006. 8, 55

[59] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2006. 1, 7, 8, 13, 55

[60] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan. WTCP: A Reliable Transport Protocol for Wireless Wide-area Networks. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 1999. 14

[61] T. Small and Z.J. Haas. The Shared Wireless Infostation Model A New Ad Hoc Networking Paradigm. In *ACM Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc)*, 2003. 12

[62] A. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2000. 7

[63] G. Tan and J. Guttag. Time-based Fairness Improves Performance in Multi-rate Wireless LANs. In *USENIX Annual Technical Conference*, 2004. 15

[64] The InterPlanetary Network. http://tmo.jpl.nasa.gov/. 13

[65] K. Uehara, H. Sunahara, and J. Murai. The InternetCAR Network Architecture: Connecting Vehicles to the Internet using IPv6. In *Conference on ITS Telecommunications (ITST)*, 2005. 9

[66] UMass DieselNet. http://prisms.cs.umass.edu/dome/index.php?page=umass-dieselnet. 9

[67] U.S. National Geodetic Survey. http://www.ngs.noaa.gov/. 21

[68] R.Y. Wang, S. Sobti, N. Garg, E. Ziskind, J. Lai, and A. Krishnamurthy. Turning the Postal System into a Generic Digital Communication Mechanism. In *ACM Special Interest Group on Data Communications Conference (SIGCOMM)*, 2004. 13

[69] T.M. Webster and J. Cardina. Accuracy of a Global Positioning System (GPS) for Weed Mapping. *Weed Technology*, 11(4):782–786, 1997. 21

[70] M.G. Wing, A. Eklund, and L.D. Kellogg. Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability. *Journal of Forestry*, 103(4):169–173, 2005. 21

[71] S.H.Y. Wong, S. Lu, H. Yang, and V. Bharghavan. Robust Rate Adaptation for 802.11 Wireless Networks. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2006. 15, 43, 55

[72] Q. Xu, R. Sengupta, and D. Jiang. Design and Analysis of Highway Safety Communication Protocol in 5.9GHz Dedicated Short Range Communication Spectrum. In *IEEE Vehicular Technology Conference (VTC)*, Spring 2003. 9

[73] S.H. Yoo, J.H. Choi, J.H. Hwang, and C. Yoo. Eliminating the Performance Anomaly of 802.11b. *Springer Lecture Notes in Computer Science*, 3421:1055–1062, 2005. 15

[74] V.C. Zandy and B. P. Miller. Reliable Network Connections. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2002. 7

[75] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad-hoc Networks. In *ACM Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc)*, 2004. 13

[76] Z. Zhuang, T.Y. Chang, R. Sivakumar, and A. Velayutham. A3: Application-Aware Acceleration for Wireless Data Networks. In *ACM Conference on Mobile Computing and Networking (MobiCom)*, 2006. 11, 38