Vehicular Opportunistic Communication Under the Microscope

David Hadaller[†], Srinivasan Keshav[†], Tim Brecht[†], Shubham Agarwal[‡]*
[†]David R. Cheriton School of Computer Science, University of Waterloo, Canada
{dthadaller,keshav,brecht}@cs.uwaterloo.ca
[‡]AirTight Networks Pvt. Ltd., Pune, India, shubham.agarwal@airtightnetworks.net

ABSTRACT

We consider the problem of providing vehicular Internet access using roadside 802.11 access points. We build on previous work in this area [18, 8, 5, 11] with an extensive experimental analysis of protocol operation at a level of detail not previously explored. We report on data gathered with four capture devices from nearly 50 experimental runs conducted with vehicles on a rural highway. Our three primary contributions are: (1) We experimentally demonstrate that, on average, *current protocols only achieve 50% of the overall throughput possible in this scenario.* In particular, even with a streamlined connection setup procedure that does not use DHCP, high packet losses early in a vehicular connection are responsible for the loss of nearly 25% of overall throughput, 15% of the time. (2) *We quantify the effects of ten problems* caused by the mechanics of existing protocols that are responsible for this throughput loss; and (3) *We recommend best practices* for using vehicular opportunistic connections. Moreover, we show that overall throughput could be significantly improved if environmental information was made available to the 802.11 MAC and to TCP. The central message in this paper is that wireless conditions in the vicinity of a roadside access point are predictable, and by exploiting this information, vehicular opportunistic access can be greatly improved.

Categories and Subject Descriptors: C.2.2 [Computer-Communications Networks]: Network Protocols

General Terms: Experimentation, Performance, Measurement

Keywords: Vehicular Communication, Opportunistic Internet Access, 802.11 MAC Bit Rate Selection

1. INTRODUCTION

As the computing power, screen size, and user interface of mobile devices evolve, users desire more services on their mobiles. As a result, technology for Internet access "on the go" has been rapidly developing, including 3G/4G cellular technology, 802.16e / WiMAX, and other MIMO-based technologies such as 802.11n. In this paper, we consider Internet access in vehicles, in particular, short-lived connections to roadside 802.11 access points that arise opportunistically as vehicles are in motion, as illustrated in Figure 1. 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 or to download product-specific promotional information from local retailers as a family drives between shopping outlets searching for a particular item; (2) **Uploading data from the vehicle**, such as for unloading digital camera images 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. [22]. In all cases, we assume the access point can act as a buffering point between the vehicle and the Internet, mitigating the effects of the backhaul link.

Previous work [18, 8, 5, 11] has confirmed the feasibility of this connectivity paradigm. In this paper, 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 existing protocols to enter a back-off state that lasts into the useful period of the connection.
- Once the useful period of the connection has begun and back-off has finished, protocols are not dynamic

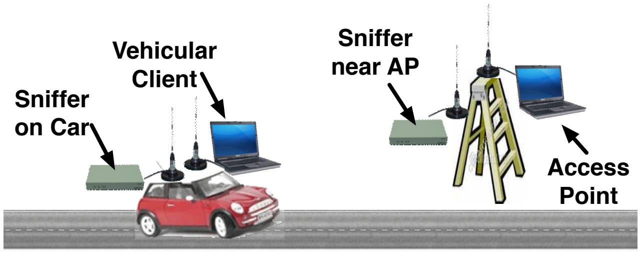


Figure 1: Our experimental setup used to analyze opportunistic vehicular connections using 802.11 hardware

[*] This work was done during an internship at the University of Waterloo

MobiSys'07, June 11-14, 2007, San Juan, Puerto Rico, USA.
Copyright 2007 ACM 978-1-59593-614-1/07/0006 ...$5.00.

enough to achieve the best possible throughput in such a rapidly changing environment.

- When leaving the coverage area, protocols again respond to wireless losses at the fringe by performing a back-off procedure, resulting in further decreases in throughput.

We demonstrate 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 known that they were in an environment where the signal strength first increases and then decreases, with reasonably predictability, they could have chosen more appropriate initial operating parameters and adjusted their adaptation mechanisms to better suit the environment. Figure 2 illustrates how specific protocol mechanisms cause undesirable effects in vehicular communication.

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 protocols in general. That is, environmental information can be used by protocols to: (1) choose better initial operating parameters and (2) tune their behavior to better handle the amount of packet loss or delay at any point in time.

Our contributions can be summarized as follows:

1. **We experimentally demonstrate that commonly used protocol stacks achieve only about half of the available potential throughput**. During a single pass of a roadside access point at highway speeds, we found that protocol behavior significantly inhibited performance.

2. **We experimentally identify a complex interplay of ten distinct causes of lost overall throughput 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.

3. **We make preliminary recommendations for best practices for using vehicular opportunistic connections**. Based on our experimental findings of how current 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.

This paper is organized as follows. After discussing related work next, we describe our experimental setup in Section 3, followed by some key observations made during our experiments in Section 4. We then show the extent to which current protocols underutilize vehicular connections in Section 5, followed by a detailed look at root causes in Section 6. Based on our experimental findings we then make preliminary recommendations for best practices for vehicular opportunistic connections in Section 7 and discuss how our work shows the broader benefits of environmental awareness for protocols in general in Section 8. Finally, we outline future work in Section 9 and then conclude in Section 10.

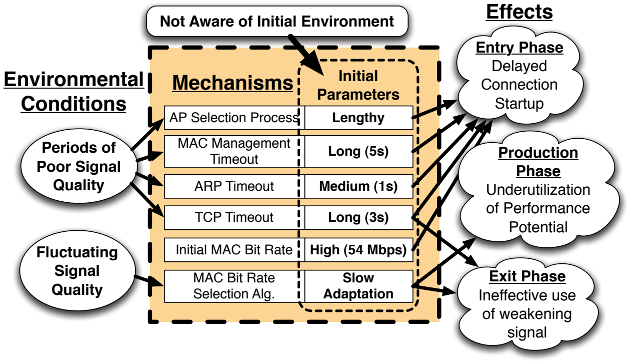


Figure 2: The cause-mechanism-effect relationship resulting in underutilization of opportunistic vehicular connections.

2. RELATED WORK

Previously, researchers have confirmed the feasibility of using opportunistic connections for vehicle to roadside communication, as well as examined the performance characteristics of such communications [18, 8, 5, 11]. In this paper, we examine characteristics of this scenario at a level deeper than previously examined and explore key open questions from prior work.

Ott and Kutscher's seminal paper proposing Drive-thru Internet [18] was the first to perform a detailed experimental analysis of opportunistic Internet access in vehicles. 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 [19], they were able to transfer 30 to 70 MB of data at 100 km/h using external antennas. They conclude that connection setup must complete before the production phase begins in order to fully utilize the connection. However, they postulate that existing protocols are not optimized for operation 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 paper.

Work by Gass et al. [8], termed *in-motion networking*, has 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 particular, they found that the multiple round trip communications required for HTTP traffic reduced total throughput by one-third compared to bulk TCP. 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. Similar poor performance due to losses of control messages during application startup was pointed out by Zhuang et al. [25].

More recently, Bychkovsky et al. [5] have conducted an extensive empirical analysis of the performance characteristics of using existing 802.11 networks for vehicular Internet

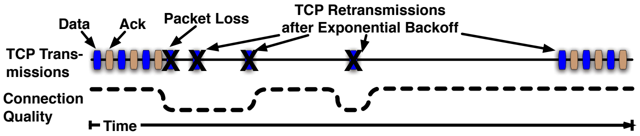


Figure 3: The well-known harmful effects of TCP misinterpreting wireless loss as network congestion (shown) have been well studied [3]. In this paper, we dig deeper into how mechanisms at both the MAC and TCP layers influence overall throughput in opportunistic vehicular communication.

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 the density of existing 802.11 access points is sufficient to support vehicular Internet through opportunistic connections.

They also investigated connection characteristics and show the mean scan, mean association, median IP acquisition, and median application initialization times to be 750 ms, 560 ms, 1.83 s, and 8s, respectively. They propose an IP address caching scheme which by-passes DHCP and reduced 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. [18] and Gass et al. [8], that high losses at the beginning of a connection could dramatically reduce the overall throughput of a connection and cite this as an area of future investigation.

In previous work, we have also confirmed the feasibility of using opportunistic connections to vehicles [11] and have shown in [10] that when multiple vehicles are in range of the same roadside access point, giving higher priority to vehicles nearer to the access point can yield throughput gains of up to four times that of standard 802.11 scheduling while still maintaining a reasonable degree of fairness.

There is a large body of work which examines the poor performance of TCP over wireless [3, 23, 2, 12] in general, such as depicted in Figure 3. However none of this work specifically considers the effects of early losses on the usability of a short-lived connection. In this paper, we do not focus on long-running steady-state connections, but rather on the effects of protocol behavior over short-lived dynamic connections, a subject not specifically explored in this related work, to our knowledge.

The Dedicated Short Range Communications group (DSRC) [6] along with the IEEE 802.11p working group are heading a safety initiative which uses short-range communication for use in the US Intelligent Transportation Systems (ITS) project. This work has the primary goal of optimizing delay-sensitive safety applications rather than the transfer of bulk data, which is the focus of this paper.

As well, achieving a useful overall connection across a series of disconnected links has been the focus of delay-tolerant networking research [22, 7, 21] in many different scenarios. However, the focus of most of this previous research has been on routing protocols, rather than on maximizing data transfer during a single opportunistic connection. Our work complements this research.

3. EXPERIMENTAL SETUP

3.1 Equipment

Our experiments involved a vehicle driving past a roadside access point at highway speeds of 80 km/h. We used two dedicated sniffers, one situated beside the access point and one on the car, depicted in Figure 1.

We used the following equipment, pictured in Figure 4:

- 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 5 foot 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.

- 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.

All hardware ran the latest release of Debian Linux testing at the time (August 2006) with kernel 2.6.16 and Madwifi driver version 0.9.1 [17].

We report on a total of 48 runs in this paper. 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 9.

Sniffer Configuration: Our goal was to ensure all frames were captured by at least one sniffer. This was not straightforward because (1) sniffer software running on either the access point or the client did not capture outgoing MAC packets, (2) dedicated sniffers placed too close to a transmitting device would not capture all packets due

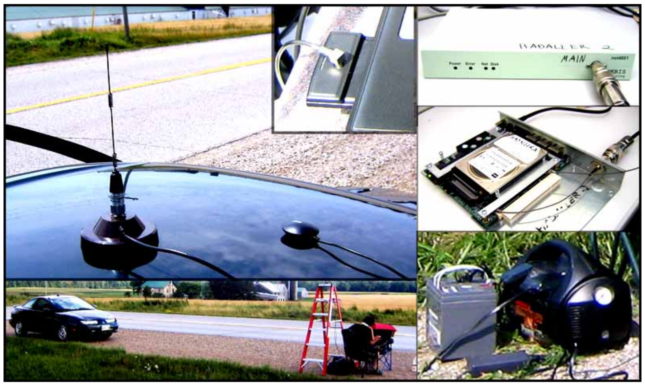
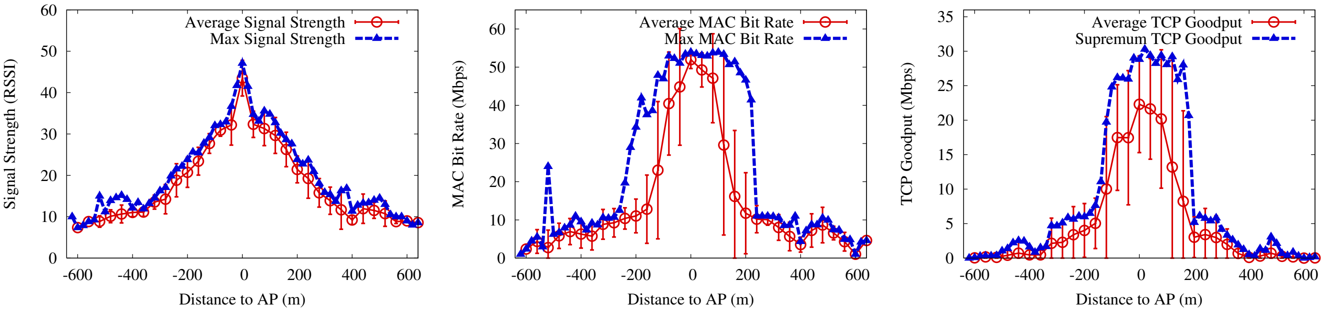


Figure 4: Equipment: Soekris board (top right), 802.11 PCMCIA card in laptop with external antenna connector (middle top), Magnetic 7 dBi Antenna and Magnetic USB GPS Receiver (left), Vehicle and roadside access point (laptop with antenna on ladder) (bottom left), and the two power units used to power the access point (bottom right).



(a) Signal Strength (b) MAC Bit Rate (c) TCP Goodput

Figure 5: 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.

to near-field antenna effects, and (3) Soekris boards were not powerful enough to run sniffing software simultaneously with transmitting or receiving data. As a result of this initial learning, we used laptops instead of Soekris boards for sending and receiving data and we used a total of four sniffing devices: two dedicated sniffers and two software sniffers on the laptops acting as the access point and client, shown in Figure 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 [20]. 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 we would have obtained similar results 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. As well, we argue that our results can be extrapolated to different vehicle speeds, as discussed in Section 9.

We used statically configured IP addresses, as DHCP is well-known to behave poorly in this environment [5], 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. We hope to explore different environments in future work, as discussed in Section 9.

3.3 Logging

Data was captured by putting 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.

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 then enters the range of the access point and continues driving at a constant speed of 80 km/h until it leaves coverage range. The experiment is then repeated in the opposite direction.

We used iperf v1.7.0 [14] 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 range, it performs a standard 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.

4. GENERAL OBSERVATIONS

While conducting these experiments, we encountered several interesting phenomena that led to further investigation. In particular we would like to highlight 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) we were able to achieve high relative accuracy of GPS measurements with some straightforward post-processing corrections for systematic GPS error.

4.1 Repeatable Connection

We found that signal strength measurements, MAC bit rate used, and TCP goodput, relative to position on the road, were relatively consistent between runs, as shown in Figure 5. Signal strength measurements in particular were extremely consistent, MAC bit rate less so due to the varying rates selected by the MAC bit rate selection algorithm. Sur-

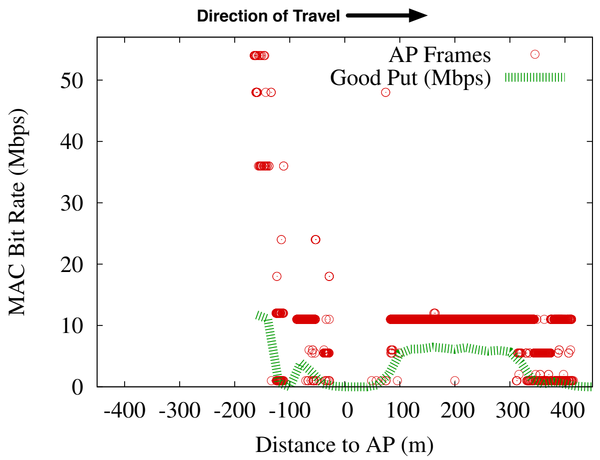


Figure 6: Example of poor rate selection using the default parameters of the default bit rate selection algorithm. Using these parameters would have resulted in 75% less overall throughput compared to using our modified parameters which achieved much higher rates (shown in Figure 5(b)). The lack of throughput during the middle of this run was due to a TCP timeout, caused by an initially excessive bit rate of 54 Mbps.

prisingly, even TCP goodput exhibited a consistent shape, even though the amount of data transferred within an individual run exhibited a fair amount of variability, as will be seen in Section 5.

4.2 Default Bit Rate Selection Unsuitable

During our initial experiments, we found that the default SAMPLE bit rate selection algorithm [4] used in the Madwifi driver was not responsive enough for our environment, and rarely selected rates higher than 11 Mbps. Figure 6 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 5(b).

In order to make use of higher bit rates we had to modify the parameters of the default bit rate algorithm to make it more responsive to the rapidly changing environmental conditions. Our modifications are shown in Table 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 reduced by 75%. Optimal choice of MAC bit rate 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 is attainable by an infrastructure provider without client modification. We experimentally explore the effects of this in Section 6.

Table 1: Changes to Default SAMPLE Rate Selection Algorithm

Parameter	**Original** [17]	**New Value**
Probe Packets	10%	40%
Sample Window	10 s	1 s
Decision Interval	Every 1000 ms	Every 100 ms

4.3 Few TCP Losses

Although the amount of MAC losses was quite high (18.5% of all transmitted MAC frames were lost), as shown in Figure 7, 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 run, there were only an average of 26.8 ± 33.3 (95% CI) losses seen by TCP ($<$ 0.13% TCP loss).

The reason for this is the 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 at most 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.

These observations call into question studies of TCP over wireless [3] that focus on losses instead of the delay-jitter due to aggressive recovery at the MAC layer.

4.4 Achieving Relative GPS Accuracy

We rely on consistent GPS measurements between different runs and accurate relative measurements between GPS devices to know 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.

While processing our data, we found two major problems with our GPS measurements: (1) The measured position of the stationary access point slowly shifted by 7.5 m over a 5.5 hour period and (2) there was a consistent 70 m difference between runs in opposite directions. The steps we took to mitigate these and other errors are explained in Table 2.

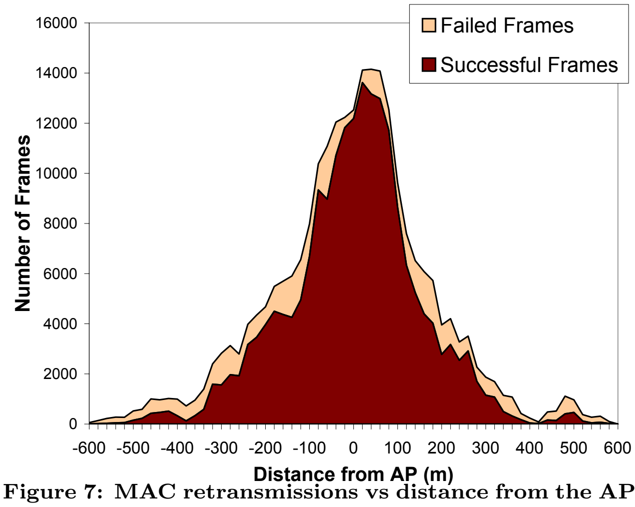
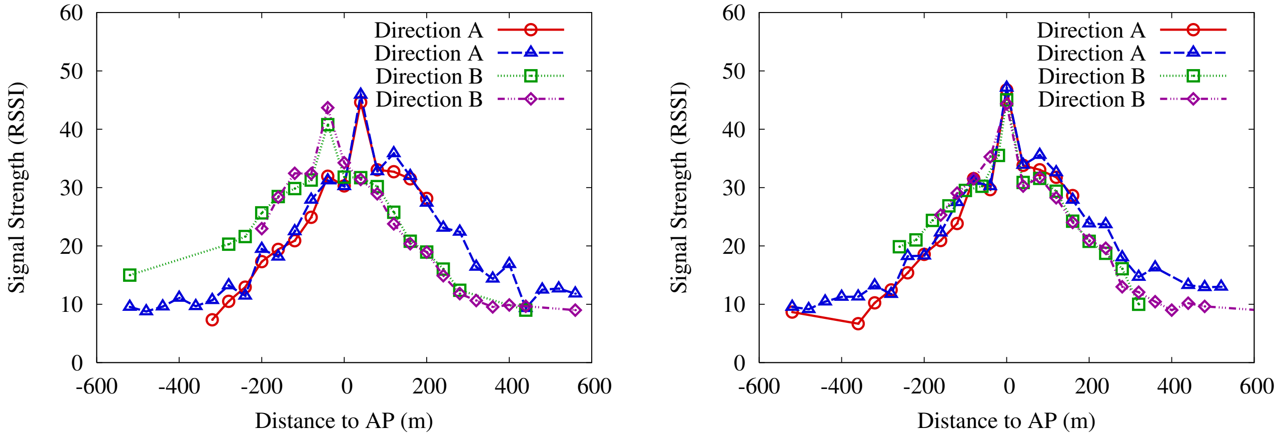


Figure 7: MAC retransmissions vs distance from the AP, aggregated across 15 runs. A higher proportion of losses occurred in the fringe areas and, on average, 18.5% of transmitted MAC frames were lost.



(a) Uncorrected

(b) Corrected

Figure 8: Correcting for GPS measurement delay, as discussed in Table 2. Signal strength measurements vs. position from four runs are shown, two in each direction. Figure A shows the 70 m position difference in measured signal peaks before correction; Figure B shows that after correcting for systematic error, the signal peaks align.

Table 2: Reducing Systematic Error

Source of Error	Impact	How we Mitigated the Error
GPS Measurement Delay: Computing the receiver's position involves sequentially measuring precise time signals from at least 3 satellites and performing trilateration. However, when the computation is performed, the measurements are already out-of-date.	3 s, at 80 km/h (67 m)	We experimentally determined the measurement delay by comparing measurements taken from runs done in opposite directions. The sharp peak of the measured signal strength was an adequate point of reference as there was a consistent separation of approximately 3 s between peaks of runs in opposite directions. Figure 8(b) illustrates that subtracting 1.5 s from the measurement time causes the two signal peaks to align.
Long-term GPS Variation: GPS measurements vary throughout the day due to atmospheric effects. GPS satellites follow a medium earth orbit, circling the earth twice each day at an altitude of roughly 20,000 km [9]. The GPS signal is delayed depending on how much of the ionosphere it travels through, which depends on the satellite's angle to the horizon. These effects are slow moving and can be tracked [9].	7.5 m drift over 5.5 hours	We measured the long term variation by comparing all GPS measurements taken at the access point throughout the day and found a maximum difference of 7.5 m and a shift of approximately 1.36 m per hour. To mitigate this slow shift throughout the day, we recalculated the access point's position for each run based on measurements taken during that run, rather than computing a global average over all measurements.
Short-term GPS Variation: We noticed that even when perfectly still, the measurements from our GPS device fluctuated. We analyzed GPS measurements taken at the access point from a sample of runs and found the average distance between any two measurements within the same run was 2 m ± 1.5 m (95% CI).	1 to 3 m	We were able to correct for this at the access point by averaging all measurements taken over a run to determine the access point's position. We could not correct for this random error in the GPS measurements taken by the vehicles while moving.
Clock drift between the client, access point, and sniffers: Ensuring close time synchronization between devices (< 50 ms) was necessary to properly correlate GPS measurements taken on different devices. We observed significant clock drift between our devices. In lab tests, we observed more than 500 ms of drift over a 24 hour period.	Up to 500 ms over 24 hours	We used NTP to regularly synchronize clocks on the two sniffers and the client with the time on the access point.
Infrequent GPS measurements: We needed to associate data frames with a GPS position, given that the GPS device only reports measurements once per second and there are often thousands of frames per second.	Up to 1 s, at 80 km/h (22 m)	To assign a finer granularity of GPS coordinates to data frames, we performed linear interpolation between measurements reported by the GPS device. A linear fit was appropriate as we were traveling in a constant direction at a constant speed. In any case, the impact of inaccurate interpolation is not significant.

5. UNDERUTILIZED VEHICULAR CONNECTIONS

In this section, we show that using existing 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 9 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 far from the median data transferred (32.6 MB), and all runs were far from their potential, explained next.

Supremum: Because no run was problem-free, Figure 9 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 5(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, 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 9, 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 [13]. Based on lab experiments we determined the TCP goodput possible for each MAC rate under ideal conditions, shown in Table 3. 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, existing protocols not only significantly underutilize connection potential by more than 50%, but also yield large variations in data transferred between runs using identical configurations. Next we experimentally analyze the causes of these problems.

Table 3: Static TCP Goodput (Lab Measurements)

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

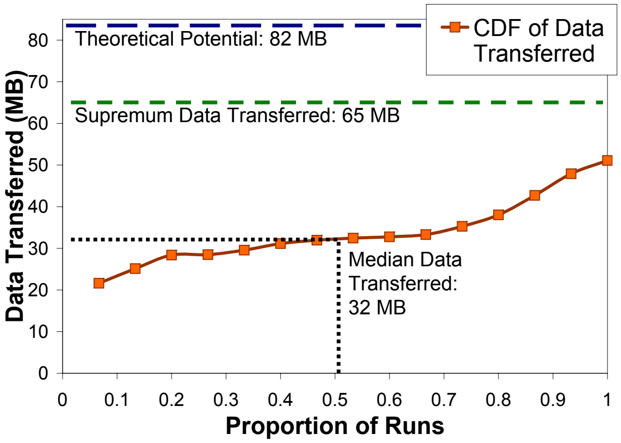


Figure 9: CDF of the amount of data transferred per run

6. CAUSES OF UNDERUTILIZATION

We have identified ten problems that occur during three distinct phases of an opportunistic vehicular connection. Table 4 provides a roadmap for this section.

Recall Figure 2, which illustrates the relationship between the observed problem areas, the mechanisms that cause them and the underlying cause: lack of environmental awareness. In this section we progress from right to left of this figure, examining each problem in detail.

6.1 Overview

For the purposes of our analysis, we divide the connection into three phases, as done by Ott et al. [18], illustrated in Figure 11. During the entry phase, connection quality is low and connection setup is performed. If connection setup completes in time, the period of good connection quality, the production phase, can be utilized. As connection quality decreases, the exit phase begins.

We have chosen to identify the production phase as beginning when 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. This results in the entry phase beginning at 640 m before the access point, the production phase between -320 m and +320 m, and the exit phase ending at 640 m past the access point. The total connection duration of the supremum is approximately 58 seconds (at 80 km/h), with the three phases lasting 14.5 s (25%), 29 s (50%), and 14.5 s (25%), respectively. Also, 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.

6.2 Entry Phase

As suggested in [18, 8, 5], 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.

Table 4: Problems causing underutilization of opportunistic vehicular connections

Observed Effect	Mechanism	Section	Figure	Impact
Entry Phase: Delayed Connection Startup (Section 6.2)	Lengthy AP Selection	Section 6.2.1	Figure 13	Average Total Delay: 13.1 s ± 12.3 (95% CI) (Figures 12, 14), resulting in a median of 7.5% lost data transferred and 22.8% in 15% of runs
	MAC Management Timeout	Section 6.2.2	Figure 15	
	Application Initialization Delay	Section 6.2.3	Figure 14	
	ARP Timeout	Section 6.2.4	Figure 16	
	Overestimation of Initial MAC Bit Rate	Section 6.2.5	Figure 17	
	Early TCP Timeouts	Section 6.2.6	Figure 16	
Production Phase: Underutilization of Performance Potential (Section 6.3)	TCP Sender: Slow Adaptation of MAC Bit Rate	Section 6.3.1	Figure 6	Goodput would have been reduced by 75% had we used the default MAC rate selection
	TCP Receiver: Slow Adaptation of MAC Bit Rate	Section 6.3.2	Figure 20	16-23% goodput reduction
Exit Phase: Inefficient Use of Weakening Signal (Section 6.4)	Overestimation of MAC Bit Rate After the Production Phase	Section 6.4.1	Figure 21	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	Section 6.4.2	Figures 21, 22	

Connection Setup Procedure

The connection setup procedure required to set up a TCP connection over an 802.11 link is shown in Figure 10. 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. [25]. 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 protocols, as shown next.

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 12 shows the locations where the connection setup completed and a TCP connection became usable, after any TCP timeouts had expired. Figure 14 shows a breakdown of the major causes of connection setup delay. We examine these causes in detail next.

6.2.1 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 13, the client would continue 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 crossed 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 [15]. Because of the repeatability of the signal strength in this environment (recall Figure 5(a)), we expected that the AP selection process would be very consistent, however, as shown in Figure 14, it was not. We attribute this to rapid fluctuations of instantaneous measured signal energy causing the average to cross the association threshold much earlier or much later in different runs.

6.2.2 MAC Management Timeout

As shown in Figure 10, once the client has decided to attempt association, it transmits a MAC authentication re-

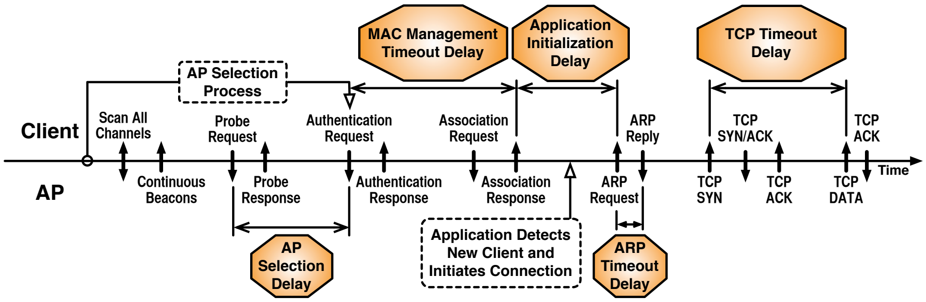


Figure 10: Timeline of connection setup procedure, with sources of delay.

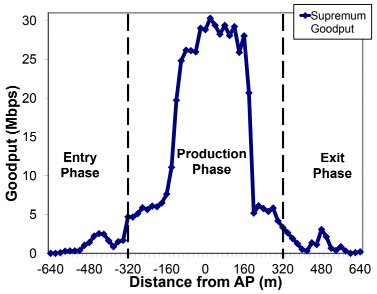


Figure 11: The three phases of a vehicular connection

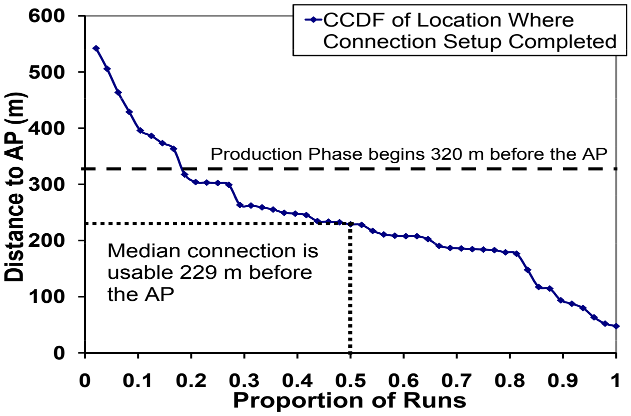


Figure 12: CCDF showing the location where the connection setup phase completed and the TCP connection became usable, indicating that very few connections made full use of the production phase.

quest. Because we are using open authentication and no encryption, this is followed by only 3 control messages to complete MAC connection setup. However, if one of these messages is lost, the Madwifi driver recovers from this loss with a **hard-coded timeout of 5 seconds** and a single retransmission. After two successive losses, the AP selection process is restarted and this process is repeated. Figure 15 shows a drastic example of how losing a few MAC management messages results in a significant delay in connection setup. Although this behavior is specific to our hardware configuration, we discuss how this observation can be generalized in Section 9.

6.2.3 *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 to DHCP and found it to be 1.8 seconds on average [5]. 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. [5] 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 14.

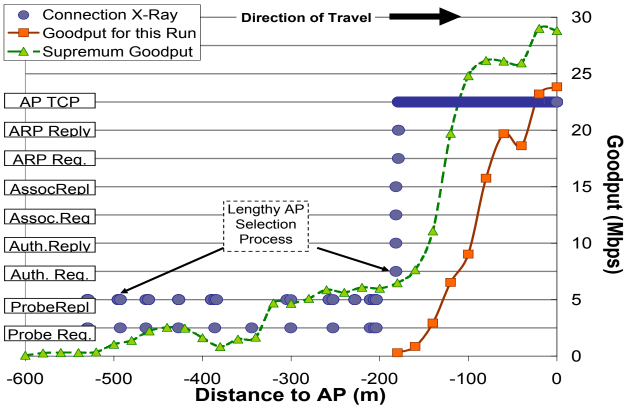


Figure 13: Example of Lengthy AP Selection

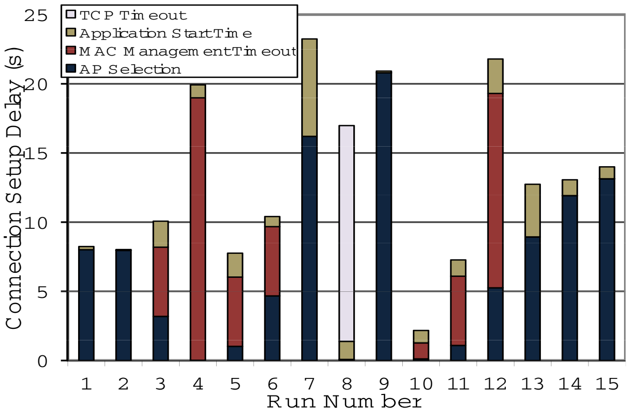


Figure 14: Cumulative bar graph showing the breakdown of the sources of delay for 15 runs

6.2.4 *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 16 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.

6.2.5 *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 using simple IF-statements based on signal strength. Analysis of our logs revealed that it was 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 5, many other algorithms do the same.

Choosing an initial bit rate that is higher than the wireless

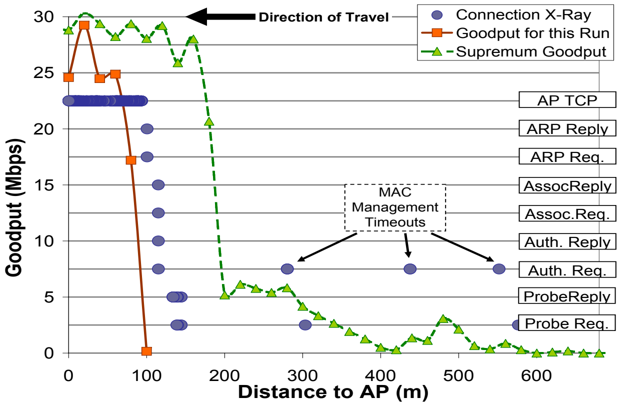


Figure 15: Example of MAC Management Timeout

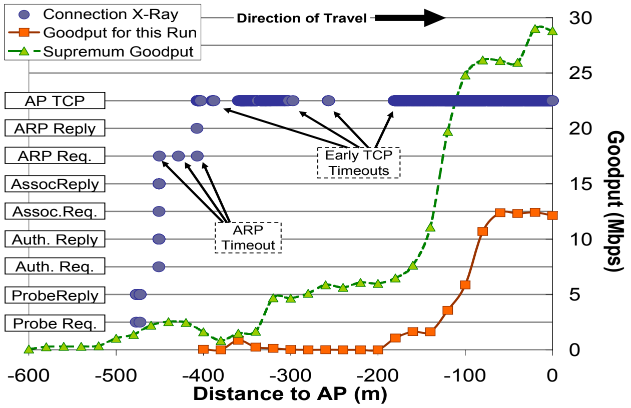


Figure 16: Example of ARP timeouts, causing 2 s delay, and early TCP timeouts, causing 10.5 s delay.

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, this lasted up to 6 seconds, as was the case in the run shown in Figure 17.

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 4.3. However, because high rates are used when lower rates would have been more suitable, the amount of viable transmission attempts is reduced and therefore high MAC losses due to overestimation of bit rate does increase the probability of TCP losses.

6.2.6 *Early TCP Timeouts*

TCP's poor performance over lossy wireless links has been well studied [3, 23, 2, 12] and found to cause problems such as that depicted previously in Figure 3. However, here we study the specific effects of TCP losses during the early phases of a connection.

When TCP loses an entire window of data, a TCP timeout occurs and TCP enters exponential back-off. During the early phase of an opportunistic connection, TCP is more likely to enter a back-off state because (a) its congestion window is smaller and (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.

While we did observe this, as shown in the run in Figure 16, early TCP timeouts were not as common as we had anticipated (recall Figure 14). 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 Atheros MAC retransmission scheme, discussed in Section 4.3. If MAC layer delays were reduced, we would expect to see an increased amount of TCP timeouts during the early phase of the connection.

Table 5: Initial MAC Rate used by Rate Selection Algorithms

Algorithm	Initial Bit Rate
SAMPLE [4]	11, 36, or 54 Mbps depending on signal strength
Onoe [17]	36 Mbps
AMRR [16]	36 Mbps
RRAA [24]	54 Mbps

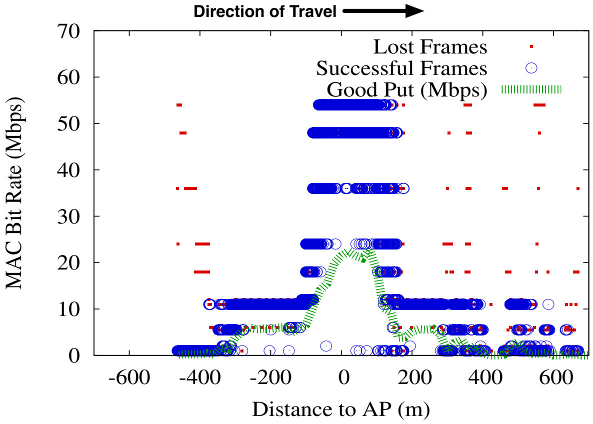


Figure 17: Example of overestimation of initial MAC bit rate; 54 Mbps was chosen although only 1 Mbps was possible during the initial phase of the connection. In this run it took 115 m (6 s) to adapt to the proper bit rate. Also seen later in the connection are the frames used to probe higher rates.

6.2.7 *Understanding the Effects of Connection Setup Delay*

Although it appears from our results (particularly Figure 14) 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 7.

6.2.8 *Impact of Connection Setup Delay*

As shown in Figure 12, 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 5(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.

6.3 Production Phase

6.3.1 *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 4.2),

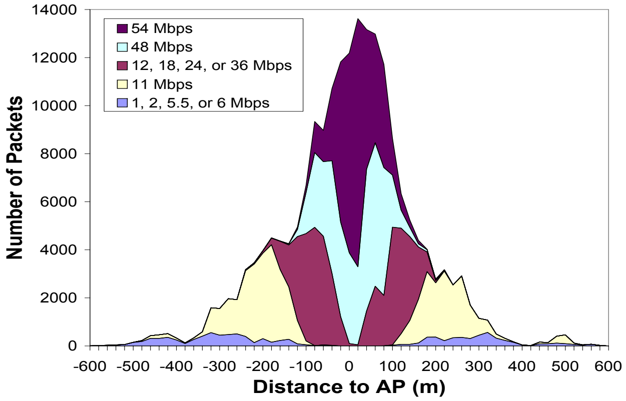


Figure 18: Breakdown of MAC bit rates used by the access point (TCP sender), aggregated across 15 vehicular runs, demonstrating the repeatability of the connection

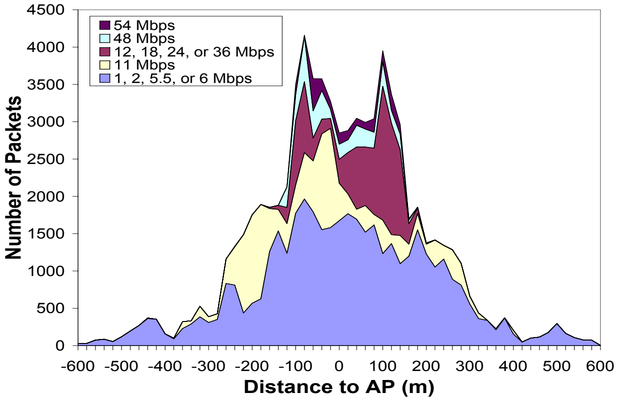


Figure 19: Breakdown of MAC bit rates used by the vehicular client for sending TCP ACKs, aggregated across 15 runs. The unusual two peaks in the curve are explained in Section 6.3.2.

requiring us to modify the algorithm's parameters to make it more dynamic for our environment.

Although the results in this paper 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.

6.3.2 *TCP Receiver: Slow Adaptation of MAC Bit Rate*

Recall from Section 3.2 that our goal in these experiments was to experimentally analyze performance characteristics 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 19) than the access point for sending TCP Data (Figure 18). 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 1). As is the case in the run shown in Figure 20, the rate was not increased until after it had passed the access point.

However, this did not occur during every run, as evident by the large error bars in Figure 5(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 20) 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 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 5(c).

Next we examine the bit rates used, by the access point and the client, aggregated across all runs, shown in Figures 18 and 19 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 6, 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.

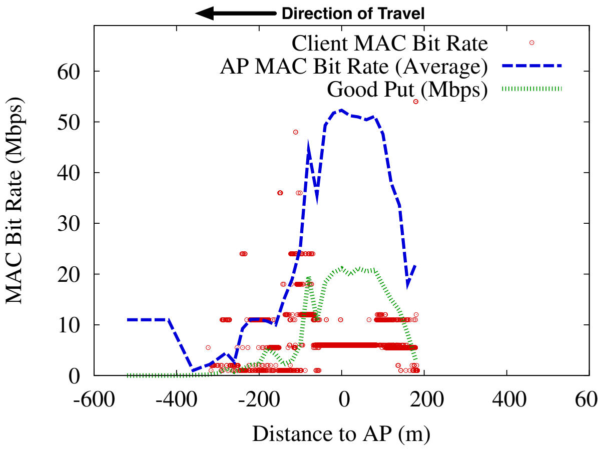


Figure 20: Example run showing slow adaptation of MAC bit rate on vehicular client (TCP receiver); the average bit rate used by the AP for this run is shown for reference

Table 6: 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.

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%

We argue that this can be reasonably extrapolated to the remainder of the production phase, and because 94% of the data was transferred during 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.

6.4 Exit Phase

6.4.1 *Overestimation of MAC Bit Rate After the Production Phase*

After the production phase, as connection quality decreased, the MAC bit rate selection algorithm on the access point, using our modified 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 6.2.5. Figure 21 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.

6.4.2 *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 also common in the exit phase. The earlier the TCP timeout occurred, the more potential data transferred was lost. Figure 21 shows an example of a TCP timeout that occurred shortly after the production phase, 300 m past the access point.

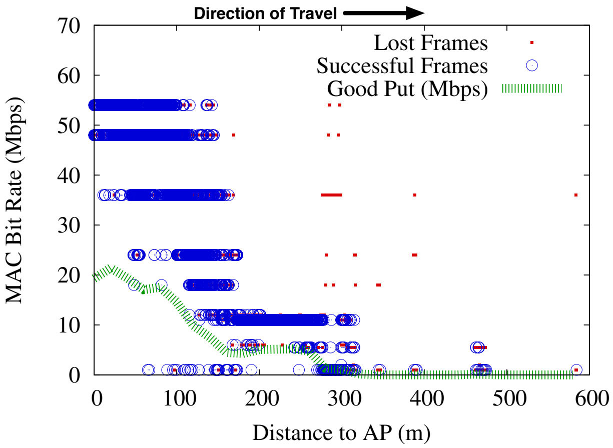


Figure 21: Example run showing overestimation of MAC bit rate after production phase causing a TCP timeout at 300 m (last half of connection shown).

Figure 22 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.

7. RECOMMENDATIONS

Based on our findings, we now suggest some best practices to other system implementers for improving vehicular opportunistic communication **using existing hardware**. In a nutshell, our recommendation is to **take whatever steps are needed to make full use of the production phase**, as this is where the majority of the data transfer occurs.

As shown in Section 6 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.

7.1 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

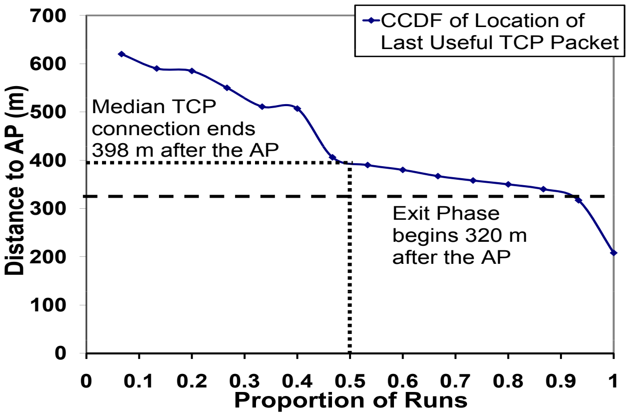


Figure 22: CCDF showing the location where the TCP connection became un-usable. As shown, the majority of runs ended early into the exit phase.

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.

7.2 MAC Bit Rate Selection Algorithms

We saw in Section 6.3.2 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 4.2).

7.3 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 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 client tuning because server-side protocols also have a significant impact on performance. In particular, the TCP sender could be any host on the Internet. 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 isn't 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.

8. DISCUSSION: THE ROLE OF ENVIRONMENTAL AWARENESS

Although the focus of this paper 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 6.2 showed that existing 802.11 MAC and TCP protocols perform poorly in marginal coverage areas. Worse, today's 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 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. If this isn't possible, they 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 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 between vehicular passes (as discussed in Section 4.1). 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 protocols and that, in the context of roadside communication, the AP can play a critical role in bringing this about. We hope to explore these insights in future work.

9. LIMITATIONS AND FUTURE WORK

The biggest limitation of our work in this paper is that we have only evaluated a single scenario: one vehicle, one vehicle speed, one environment, one wireless card, and only downlink data transfers. However, some of these aspects of opportunistic connection have already been examined. Previous work [18, 8, 11] has shown that data transferred is roughly proportional to vehicle speed. As well, in [11], we examined data transfer in the uplink direction and found similar behavior to that in the downlink direction. Additionally, we are currently studying the impact of multiple vehicles on opportunistic connections. Preliminary findings indicate that if two vehicles sequentially pass an access point, the first vehicle's data transmissions significantly interfere with the delivery of the second vehicle's MAC control messages, delaying the second vehicle's connection setup. Bychkovsky et al. [5] have studied vehicular opportunistic connections in an urban setting, however further exploration is needed to understand the details of MAC layer behavior.

Gaining a thorough understanding of the behavior of different wireless cards and different MAC rate selection algorithms is an important area of future study. However, the lack of mechanisms in the 802.11 standard to allow adaptation to the vehicular environment will affect every card. Therefore, design tradeoffs will necessarily have been made by vendors to optimize their cards for some environment and not others. Furthermore, performance problems attributed to TCP are card-independent.

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? We hope to explore these questions in future research.

A higher layer problem, which has been a particular challenge for delay tolerant networking research [7, 22, 21], 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.

10. CONCLUSION

In this paper, we experimentally analyze protocol interaction for opportunistic vehicular communication at a depth not previously examined. We demonstrate that heightened awareness of the operating environment, particularly in the vehicular scenario, can dramatically increase the overall throughput of a connection.

In particular: (1) We found that in our experiments, current protocols only transfer an average of 50% of the data possible in this scenario. Specifically, we show that losses during the connection setup phase contribute significantly to this amount. (2) We have identified ten problems that cause this throughput reduction, including: (a) *Entry phase:* 7.5% (median) to 22.8% (15th percentile) reduced data transferred due to a lengthy AP selection process, high MAC management timeouts, ARP timeouts, overestimation of the initial MAC bit rate, and TCP timeouts early in the entry phase, (b) *Production phase:* Up to 75% less data would have been transferred if we used the default MAC bit rate selection algorithm in our card's driver on the access point, and 16 to 23% less data was transferred due to slow adaptation of MAC bit rate on the client during the production phase, and (c) *Exit phase:* 2.1% (median) to 3.0% (15th percentile) less data was transferred due to overestimation of MAC bit rate and TCP timeouts early in the exit phase.

We further suggest best practices for vehicular opportunistic connections: (1) We argue that the best way to use current protocols in this scenario is to avoid the fringe areas altogether, due to protocol timeouts and back-off procedures extending into the production phase, and (2) 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 2 by using such environmental information.

ACKNOWLEDGEMENTS

We would like to thank Kimberly Tsang for her assistance running experiments. We also gratefully acknowledge the Natural Sciences and Engineering Research Council of Canada for financial support for portions of this project.

11. REFERENCES

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

[2] H. Balakrishnan and R. Katz. Explicit Loss Notification and Wireless Web Performance. In *IEEE GLOBECOM*, 1998.

[3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking*, 5(6):756–769, 1997.

[4] J. Bicket. Bit-rate selection in wireless networks. Master's thesis, MIT, 2005.

[5] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *ACM MobiCom*, 2006.

[6] Dedicated Short Range Communication group. http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[7] DTN Research Group (DTNRG). http://www.dtnrg.org/.

[8] 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)*, April 2006.

[9] GPS SPS Signal Specification, 2nd Edition (June 2, 1995).

[10] 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.

[11] D. Hadaller, H. Li, and L. G. A. Sung. Drive By Downloads: Studying Characteristics of Opportunistic Connections. In *USENIX NSDI Poster Session*, 2005.

[12] H.-Y. Hsieh, K.-H. Kim, Y. Zhu, and R. Sivakumar. A receiver-centric transport protocol for mobile hosts with heterogeneous wireless interfaces. In *MobiCom*, 2003.

[13] IEEE 802.11a-1999 (8802-11:1999/Amd 1:2000(E)).

[14] Iperf. http://dast.nlanr.net/Projects/Iperf/.

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

[16] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *ACM MSWiM*, 2004.

[17] Multiband Atheros Driver for WIFI. http://www.madwifi.org/.

[18] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for Automobile Users. In *IEEE Infocom*, 2004.

[19] J. Ott and D. Kutscher. Towards Automated Authentication for Mobile Users in WLAN Hot-Spots. In *IEEE VTC*, Fall 2005.

[20] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based Characterization of 802.11 in a Hotspot Setting. In *ACM SIGCOMM E-WIND Workshop*, 2005.

[21] 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.

[22] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In *MobiCom*, 2006.

[23] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan. WTCP: a reliable transport protocol for wireless wide-area networks. In *MobiCom*, 1999.

[24] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MobiCom*, 2006.

[25] Z. Zhuang, T.-Y. Chang, R. Sivakumar, and A. Velayutham. A3: Application-Aware Acceleration for Wireless Data Networks. In *MobiCom*, 2006.