Measurements of In-Motion 802.11 Networking

(Richard Gass[†], James Scott[†] and Christophe Diot[‡]
*Intel Research[†], Thomson R & D[‡])

IRC-TR-05-050

Research at Intel

Measurements of In-Motion 802.11 Networking

Richard Gass†, James Scott†, and Christophe Diot‡
†Intel Research, ‡Thomson R&D

***Abstract*— Wireless networking can support in-motion users by providing occasional opportunities to transmit and receive data. We measure the performance of UDP and TCP transfers between a car traveling at speeds from 5 mph to 75 mph, and an 802.11b access point. We analyze the impact of bandwidth and delay limitations in the backhaul network on the feasibility of in-motion transfer with typical Internet applications. We observe that in interference-free environments, a significant amount of data can be transferred using off-the-shelf equipment. We find that performance suffers mostly from network or application related problems instead of wireless link issues, i.e., protocols with handshakes, bandwidth limitations, and long round-trip times.**

I. Introduction

Wireless networking, and in particular IEEE 802.11, has recently become popular for home networking and Internet access in so-called "hot-spots" (e.g., airports, cafes, and restaurants). However, the technology has been designed for static utilization, nothing prevents it to be utilized for in-motion scenarios.

Before considering extending Internet legacy services to in-motion users, it is worth studying experimentally the 802.11 technology capabilities to support such applications while on the move. It is well known that TCP/IP-based applications do not work well under conditions where network connectivity is transient and where link quality is highly variable. The performance of TCP in wireless network scenarios has been well-studied [1]. Many research efforts have attempted to mitigate these problems.

Our motivation is to understand the factors limiting the performance of wireless networks for in-motion users using realistic applications. In other words, how well can a user moving past an access point, e.g., in a car or bus, make use of these transient opportunities to perform such tasks as browse the web, send and receive email messages, or conduct file transfers.

We take an experimental approach to the problem and present measurements from a study of 802.11b networking involving a user in a moving car and a wireless access point (AP). These measurements have been realized in the California desert, a radio interference-free environment (i.e., in absence of 802.11 or other conventional wireless networking signals), in order to be able to understand what networking parameter caused the performance observed. The presence of radio noise would be realistic but it would also limit our abilities to perform root cause analysis.

Firstly, we have analyzed the performance of TCP bulk, UDP (at various packet sizes), and web traffic for in-motion transfers between a mobile device and an AP. In the second phase of experiments, we introduced a backhaul network segment between the wireless AP and server, thus emulating more realistic Internet characteristics, to study the impact of long delay and bandwidth bottlenecks on the performance of two very popular Internet applications: file transfer and web browsing.

At this early stage of data analysis, we have been able to (i) show that it is possible to communicate with an 802.11b AP while moving, (ii) confirm and deepen experimental results obtained by other researchers, and (iii) identify the factors that limit the performance of opportunistic communication. This paper contributes to the understanding of in-motion wireless transmission in multiple ways. First by being experimental and radio-interference free, it provides realistic upper bounds on the performance of in-motion data transfer with 802.11. Second, it helps us understand what changes or adaptation need to be made to the current networking technology to provide a new range of communication services.

This paper is structured as follows. We first present related work in vehicular WiFi measurements. We then describe our experimental methodology, followed by results. We discuss the most significant limiting factors experienced by in-motion users and propose a few solutions to these issues.

II. Related Work

In 2002, Singh et al. reported on wireless link properties between two moving cars with externally mounted antennae [3], using UDP traffic to focus on the wireless link properties. In 2004, Ott and Kutscher described TCP and UDP measurements between a moving car with an external antenna and an AP [2]. Their work explores TCP and UDP traffic with the server being directly connected to the AP.

Our work distinguishes itself from previous work (and from [2] in particular) in a number of ways. We show that internal laptop antennae are able to support in-motion wireless networking with good performance. We examine the wireless network range, i.e., the time and distance when the client can associate, and the actual usable range. Also, we analyze the feasibility of using current applications for in-motion scenarios. This leads us to investigate two specific factors, which are not examined by previous work, the realistic application traffic and the effects of the backhaul network segment. Last, we also performed a significantly higher number of tests than previous experimental studies, thus obtaining higher confidence on the reported results.

III. Experimental Design

We chose to focus on the effects of motion and short connection opportunities on Internet protocols and applications, and we are less interested in physical layer effects. We therefore performed these experiments in a situation where the physical channel was uniformly behaved in space and over time, namely on a straight, flat, and traffic free road in the California desert. These conditions meant that:

- Successive runs of the same experiment produced very similar results. This is necessary given the limited number of experiments we could perform and the high numbers of parameters to try. The consistency among multiple experiments is mandatory to validate our observations.
- We could study the effects of a single network variable without the presence of radio interference, eliminating unknown observations and easily isolating root cause phenomena.

A. *Experimental parameters*

We varied three major parameters in our experiments: car speed, network traffic type, and backhaul network performance. These parameters are now discussed.

We used six car speeds: 5, 15, 25, 35, 55 and 75 mph. This target speed served as a guide for the driver. We also measured the average speed by timing the car over the measured test range distance.

We explored three traffic types, namely UDP (with various packets sizes), TCP bulk traffic (simulating FTP file transfers), and web traffic (HTTP over TCP). To test UDP, we used a stream of data consisting of successive UDP packets with sizes 50, 100, 200, 400, 800 and 1500 bytes. This allowed us to study the effects of packet size on in-motion wireless transfers. For the TCP bulk traffic, a stream of data was sent using 1500 byte packets. For web traffic, we used the Apache web server with a cache of 6 popular websites[1] (including embedded objects), and used a web crawler to download the web pages (again including embedded objects) in a cycle.

We also simulated the effect of the backhaul network, i.e., the network path between the AP and the server. We tested two parameters: a 1 Mbit/s bandwidth limit (simulating a typical DSL access network), and a 100 ms latency each way (simulating an inter-continental network path). We tested these effects both separately and together.

Given a finite time for testing, there is an obvious trade-off between the variables that can be explored and the number of repetitions of each test. We chose to repeat each test just two times in order to perform the full exploration of the above variables, giving us a total of 108 tests. These tests took three, twelve hour days of work in the desert for three people. Note that consistency of the two similar experiments can be checked against other experiments with similar parameters. Therefore, it is important to note that a given parameter is not only measured twice.

B. *Hardware and software*

The experimental setup is illustrated in Figure 1. The client is an IBM Thinkpad T41 with integrated Intel Pro/Wireless 2915ABG wireless adaptor. The client communicates with a Linksys WAP55AG access point operating on channel 1 in 802.11b mode only. The APs WAN port is wired to a laptop modeling the backhaul network. That laptop featured an IBM Thinkpad T30 with one built-in ethernet adaptor and one PCMCIA ethernet adaptor. The T30's other ethernet adaptor is connected to the server, an IBM Thinkpad T42. Both the client and server are also equipped with a PCMCIA Netgear WAG511 802.11 wireless adaptor used for network monitoring only. We considered using DHCP, but Ott and Kutscher's analysis [2] showed highly variable performance due to slow retransmission timers. Therefore, we decided to use preset IP addresses for all machines.

[1]The websites used were BBC, CNN, Google, Intel, Yahoo, and Slashdot. The average page size was 76KB.

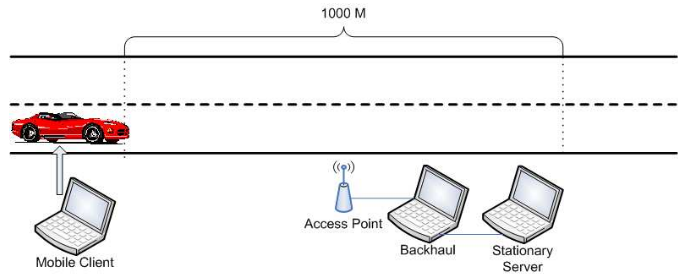


Fig. 1. Setup for experiments

The hardware above was chosen to be typical of that used by real users. No external high-gain antennae were used on the clients or the access point, and default link layer parameters were always used.

The software configuration is as follows. All machines used the Fedora Core 3 Linux distribution, with the default 2.6 Linux kernel. UDP traffic is generated using a `bash` shell script and the `/dev/udp` interface. `iperf` is used to generate the TCP bulk traffic, while the web traffic is generated using the `wget` program on the client and Apache `httpd` on the server. The backhaul computer used the Linux `tc` program with the `netem` module to implement the bandwidth and delay models.

Experimental data is logged in two ways. We generate a `tcpdump` log on the active network interfaces of both the client and the server. In addition, we used the `kismet` wireless network monitoring tool at the client and the server to obtain link-layer traces.

C. *Experimental conditions*

The AP is placed at the side of the road on a 160 cm tripod. A GPS unit with reported accuracy of two meters was used to mark the road 500 m (well out of AP range) either side of the AP with paint and signs. Each test begins with the client computer on the lap of the passenger in the car well beyond the 500 m distance, at which point the test scripts are started and the car comes up to and maintains the target speed. As it passes the first 500 m mark, the enter key is pressed on the client causing a timestamp to be recorded. When the client comes into range it automatically associates and begins sending test traffic. As the car passes the end 500 m mark, the enter key is pressed again, causing another timestamp to be recorded. We chose to use keypresses for timing instead of

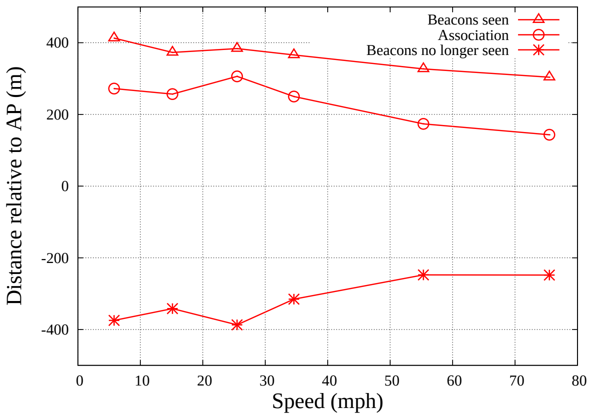


Fig. 2. Wireless network range

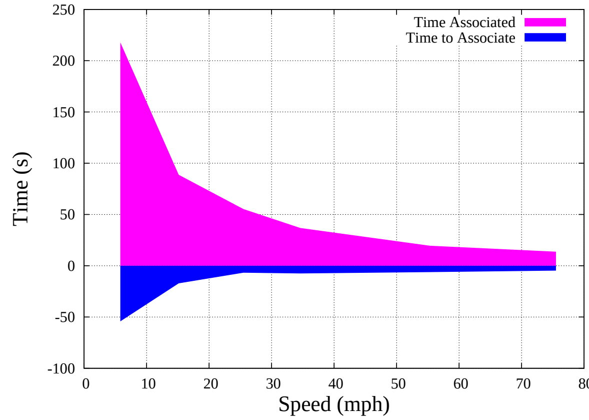


Fig. 3. Association Time

GPS for convenience and because the operator can clearly see the sign approaching and time the keypress accurately enough. Even if GPS was used, it would not guarantee the accuracy of the measurements because of the two meter reported accuracy in our unit.

IV. Wireless Link Performance

We first analyze the raw performance of the wireless channel under in-motion conditions. This gives us a base case against which we can compare the performance when using realistic traffic types and when using a backhaul network (in the following sections). In this set of experiments, the mobile device communicates directly with the fixed AP. Therefore, we expect minimum delays and very low loss when the two devices are in range.

A. Wireless range

Figures 2 and 3 shows the association range relative to the AP and association duration respectively, at various speeds. Data from all traffic types and repetitions were used to provide average actual speed and range values for the set of experiments at each particular target speed (based on a total of 108 experiments).

In Figure 2, the three lines represent the client's position when it first receives the APs association beacon (top line), when it successfully associated with the AP (middle line), and when the final beacon was received by the client (bottom line). As the graph shows, the client is able to see beacons from 412 m to 303 m away at 5 to 75 mph respectively, but does not successfully associate until 272 m to 143 m away. Some of this delay is due to the time taken to perform link-layer association, while some is due to the fact that the client can successfully decode packets at a range beyond that which it can actually successfully transmit. As the speed increases, the distance for which the client is in range decreases, from 272 m at 5 mph to 143 m at 75 mph. However, even at 75 mph there is still a 392 m window for data transmission, which represents a usable connection opportunity for moving clients.

Figure 3 illustrates the association overhead and usable data transfer window at various speeds. The association overhead remains constant when the speed is greater than 25 mph which shows that once a client is in range of the AP, the speed of the mobile client has little to no effect on the time required for association. The area of the graph that is above zero represents the amount of time where data transmissions can occur. Comparing the distance to the time in range, the effective distance window size does not vary much with speed (Figure 2), but the time spent in that window reduces significantly, from 217 seconds at 5 mph to only 12 seconds at 75 mph (Figure 3). However, as it will be seen later in this paper, the amount of data that can be successfully transmitted is still significant.

B. Packet losses

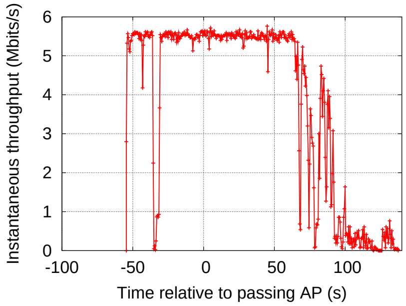
When investigating packet loss against speed, our interest is whether higher speeds might cause more loss, and in particular, cause greater losses for larger sized packets due to the more rapidly changing channel conditions. We therefore experimented using UDP traffic with various packet sizes at different speeds as shown in Figure 4.

This graph shows that loss rates increase threefold from 5 mph to 55 mph, but there is no significant effect on packet loss. At 75 mph, the loss is approximately doubled again, with some indication (at low packet sizes) that packet size may have a minor effect. However, the overall loss rate is well below 1%.

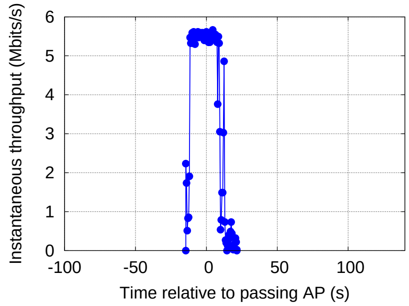
Further examination of the raw logs shows that losses only occur when the mobile client is at the limit of the wireless network range; once inside a nominal range of 150 m, there are no losses occurring for all speeds and packet sizes. We also examined the loss rate for traffic from the server to the mobile client. In this direction, there is loss even when in the nominal 150 m range, however, the rate never exceeds 1%. When compared to loss rates in deployed wireless networks, this is negligible [5].

C. Instantaneous throughput

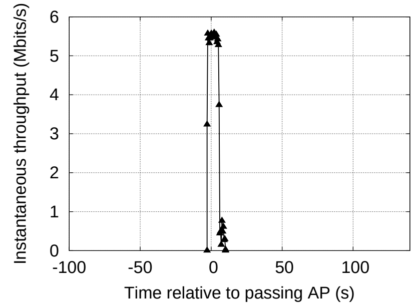
Figure 5 shows the instantaneous throughput across time over 500 ms intervals for various speeds and traffic types. UDP and TCP bulk results both show that the channel goes through an initial phase of varying performance before a stable well-behaved phase while in good range, with a final period of



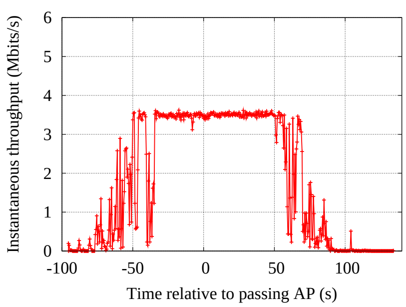
(a) TCP bulk traffic 05 mph



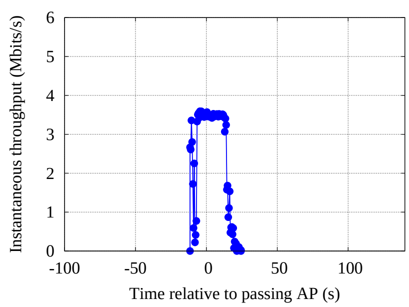
(b) TCP bulk traffic 35 mph



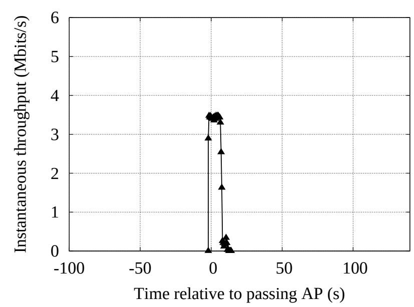
(c) TCP bulk traffic 75 mph



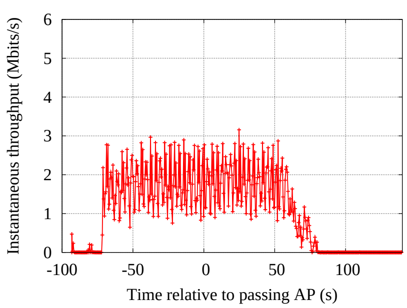
(d) UDP traffic 05 mph



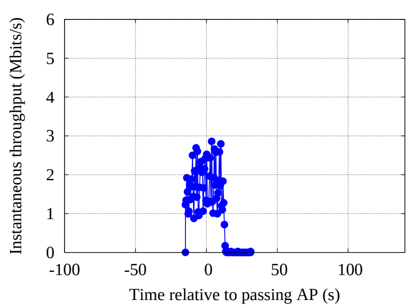
(e) UDP traffic 35 mph



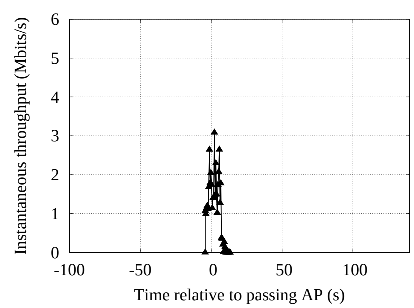
(f) UDP traffic 75 mph



(g) Web traffic 05 mph



(h) Web traffic 35 mph



(i) Web traffic 75 mph

Fig. 5. Individual instantaneous throughput for various speeds and traffic types

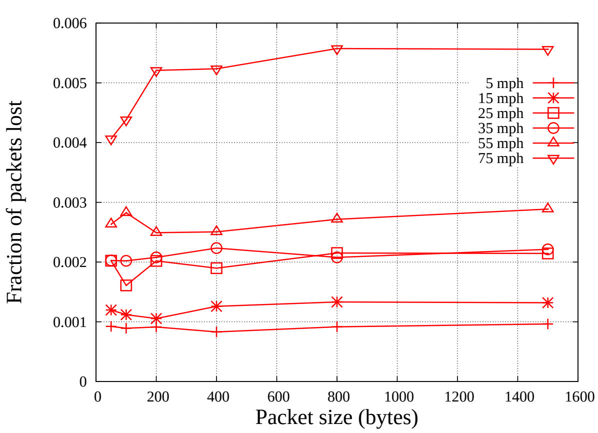


Fig. 4. UDP Packet loss

varied performance as the client moves out of range. This is what Ott and Kutscher [2] named the entry, production, and exit phases. Interestingly, at 75 mph the entry and exit phases seem to disappear, while the production phase remains highly visible. This is a very encouraging result for the usability of transient connection opportunities even at higher speeds. We conjecture that at high speed, the entry and the exit phases are shorter (but more lossy) and that the mobile almost “skips” this transition zone to enter the production phase. This can be explained by the fact that beacons can be exchanged before the two devices are in transmission range.

TCP bulk traffic clearly exploits the channel best, with an achieved bandwidth of 5.5 Mbit/s during the production phase, in all tests from 5 mph to 75 mph. Despite its handshake protocol, TCP does not seem to impact the duration of the production phase (as compared to UDP).

UDP traffic achieved a lower throughput of 3.5 Mbit/s. This

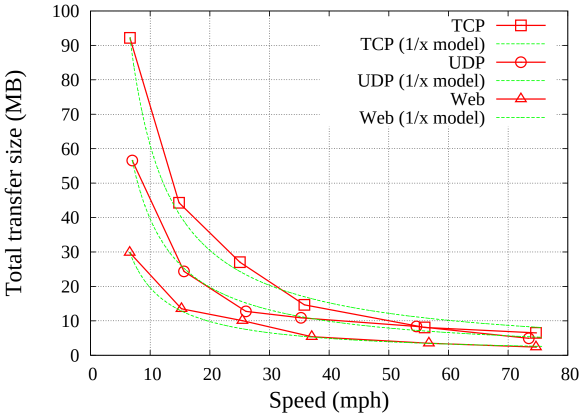
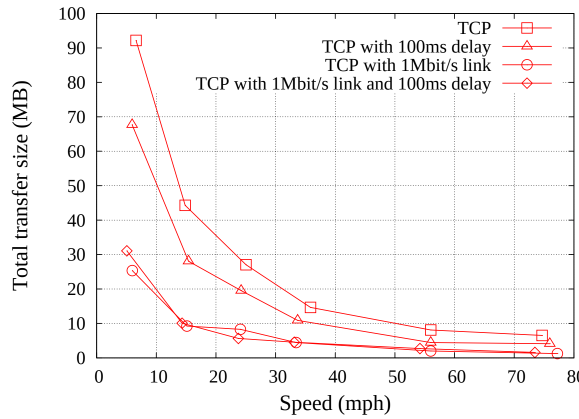
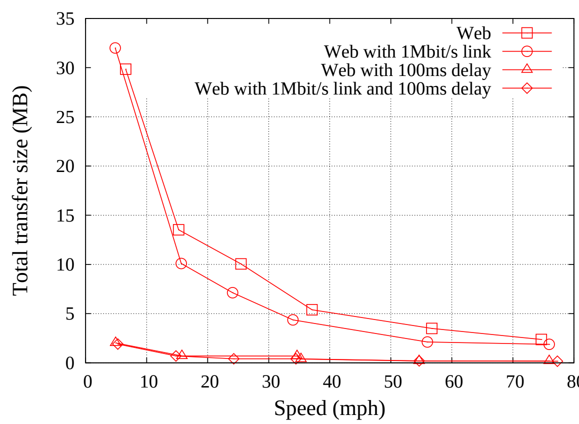


Fig. 6. Total data transferred at various speeds



(a) TCP bulk traffic



(b) web traffic

Fig. 7. Total data transferred at various speeds, with constrained backhaul networks emulator

is due to the fact UDP packet sizes range from 50 bytes to 1500 bytes, which results in the link layer overhead being higher compared to the 1500 byte TCP traffic.

Web traffic shows a lower and much more variable transfer rate, with an average of around 1.8 Mbit/s during the production phase. This is due in part to variable packet sizes. However, the main cause is that web traffic uses an application-layer protocol that introduces delays that are not due to the network itself. This is significant, because while many tests of wireless networks are conducted on canned traffic, real life traffic has application-layer dependencies and will therefore not necessarily make optimal use of the available channel. We will see later in this paper that the impact of these application level protocols is even more dramatic in the presence of a backhaul network.

D. *Total data transferred*

Figure 6 shows the total amount of data transferred at various speeds. UDP, TCP file transfer, and web traffic make different use of the channel as described previously, resulting in different total transfer sizes. To help gauge these curves, a $1/x$ graph is plotted for each, which extrapolates the performance from the 5 mph result, assuming that the only effect of higher speeds is a lower available connection time (i.e., 5 times less for 25 mph). The actual and $1/x$ lines track each other closely for all three traffic types, showing that higher speeds are not affecting the quality of the wireless network significantly (at least in the production phase), which is an encouraging result for applications making use of in-motion 802.11 connections.

In order to illustrate what is possible with in-motion transfer over a wireless link, we summarize the various parameters as follows with real life examples. At fast paced walking speed (5 mph), a 92 MB file can be transferred with TCP while passing by an AP. Most Internet legacy applications would therefore perform pretty well. At 75 mph, most applications would be in trouble, unless they are modified to perform bulk data transfer. However, a significant amount (6.5 MB) of data can be transferred during the production window which would be enough to download news or relevant traffic information, send and receive email, or even download an mp3 file.

V. Backhaul parameters

We now analyze the impact of backhaul network parameters for in-motion wireless transfers on TCP bulk and web traffic.

Figure 7(a) shows the effects of the backhaul network which we defined in Section III-A for TCP bulk traffic. The additional delay has little impact, which is predictable since TCPs transmit window allows many packets to be present in the network at once. However, the bandwidth limitation has a much larger impact. With a 1 Mbit/s link on the backhaul network, the data transferred reduces from 92 MB to 25 MB at 5 mph and 6.5 MB to 1.2 MB at 75 mph.

Figure 7(b) shows very different effects for web traffic. In this case, it is the added delay which has the most serious impact on the total throughput, since the increased time penalty of the end-to-end HTTP protocol of requests and responses causes the channel to remain sorely under-utilized. The added delay causes a drop in total data transferred from 29 MB to 2 MB at 5 mph and 2.3 MB to 187 KB at 75 mph.

The bandwidth limit, on the other hand, causes throughput to drop only slightly (as the amount of data transferred was already reduced at the application-level). This can be further illustrated by comparing the 1 Mbit/s limit in Figure 7(b), with the 1.8 Mbit/s utilization that web traffic exhibited in the non-limited case for the instantaneous web traffic in Figure 5.

Backhaul constraints will always be present in real-life scenarios and, as our results show, will have a negative effect on the amount of data transferred. However, the backhaul constraints do not have to cause as large a decrease in throughput as in Figure 7(b). Modifications to heavy application level communication protocols such as HTTP might benefit in-motion wireless transfers by reducing the number of end-to-end round trips in the critical path of data transfers, greatly improving the achievable throughput.

VI. Consequences for in-motion data transfers

The results above have shown that there are multiple limiting application and user level factors affecting in-motion wireless connection opportunities, but that the wireless technology itself is not one of them.

One issue for in-motion users is the limited connection window. The amount of time that a user can use a transient connection decreases significantly with speed. Most applications require the user to be involved with connecting and authenticating to services (e.g., VPNs, websites, email servers, etc.), a process which may, in itself, dominate or exceed the available time of a transient connection. Therefore, a mechanism to automate the process of connecting and authenticating would be essential for in-motion usage.

We also observed that poor performance was caused by application level protocols with multiple request cycles leading to the under-use of transfer opportunities. Our results show that web traffic consumed only one third of the bandwidth that TCP bulk traffic consumed, and that latency in the backhaul network makes this effect much worse. This is due to the multiple transactions needed to complete most web page requests, and to the multi-round-trip nature of application protocols such as HTTP.

Our second recommendation is to avoid this situation by reducing the number of network round trip times required for user applications as much as possible. For the web application, this could be achieved by allowing a client to request a web page inclusive of embedded objects, thereby requiring only a single request and single response[2].

At present, wireless hot-spots are typically used by stationary or nomadic users, and not by in-motion users with transient connectivity. While the recommendations above address the optimization of the performance of such transient opportunities, the wider issue is that the majority of current users do not have devices which can make use of these opportunities. A laptop must be in standby mode while being carried or it will overheat in its case. A PDA does not have a heat problem, but normally performs aggressive power saving, turning wireless channels off while not actively used by its owner.

Our final recommendation is that mobile users must have at least one device which is designed for always-on, low power activity. Such a device could perform communications during transient connections and might later provide data to other devices which were in standby. This device could be a stand-alone device, or be integrated into another device carried by a user. Candidate devices include mobile phones, which have similar power profiles and are beginning to be built with 802.11 capabilities, or a dedicated device [4].

[2] This is different from HTTP 1.1 pipelining, which requires one connection to be opened to each of the many servers that the embedded objects are typically spread across.

VII. Conclusions

We have experimentally demonstrated the feasibility of using 802.11 networking, with standard antennas and link-layer parameters, for networking between an access point (AP) and a device which is moving past an AP. By conducting a study in an interference-free environment, we provide a clear picture of the 802.11 channel for in-motion scenarios based on 108 drives past an AP. We show that, for the entire speed range from 5 mph to 75 mph, there is a region of good connectivity during which loss rates are low and throughput is high. However, for application-layer protocols such as HTTP, we found that this throughput is not achieved since multiple round-trip dependencies exist and lead to unused capacity. This situation is greatly worsened when realistic access network parameters are present, particularly due to the effect of end-to-end delay when combined with round-trip dependencies.

We make three recommendations to improve the utilization of transient connectivity for in-motion 802.11 users. First, the numerous slow or manual authentication and authorization stages involved in today's 802.11 networks must be eliminated. Second, we suggest the introduction of "bulk modes" for protocols such as HTTP, in which a single request-response round-trip might be used to create a bulk transfer of data which is currently sent over many round-trips. Finally, we advocate an always active device that can negotiate connections on the client side (requiring no changes in existing infrastructure equipment), while allowing updates to data (e.g., email, web cache, etc.) when passing transient connection opportunities.

There is plenty of work to be accomplished in this area. More measurements of transient channels, e.g., between two vehicles, or in different physical environments, would facilitate better understanding of the wider scenario. Automated and streamlined authentication methods for 802.11 access networks and for online applications must be deployed. "Bulk modes" for protocols need to be proposed and evaluated. Our work shows that, with such efforts, it might be possible to employ transient, in-motion connectivity to satisfy users' needs.

References

[1] Balakrishnan, H., Padmanabhan, V. N., Seshan, S., and Katz, R. H. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking* (December 1997).

[2] Ott, J., and Kutscher, D. Drive-thru internet: IEEE 802.11b for automobile users. In *IEEE INFOCOM* (2004).

[3] Singh, J. P., Bambos, N., Srinivasan, B., and Clawin, D. Wireless lan performance under varied stress conditions in vehicular traffic scenarios. In *IEEE Vehicular Technology Conference* (2002).

[4] Yaap: Yet another access point. http://www.cambridge.intel-research.net/∼rgass/yaap.

[5] Yeo, J., Banerjee, S., and Agrawala, A. Measuring traffic on the wireless medium: Experience and pitfalls. Technical Report CS-TR 4421, Department of Computer Science, University of Maryland, December, 2002.