Eliminating handoff latencies in 802.11 WLANs using Multiple Radios: Applications, Experience, and Evaluation

Vladimir Brik, Arunesh Mishra, Suman Banerjee *

ABSTRACT

Deployment of Voice-over IP (VoIP) and other real-time streaming applications have been somewhat limited in WLANs today, partially because of the high handoff latencies experienced by mobile users. Our goal in this work is to virtually eliminate this latency by exploiting the potential of multiple radios in WLAN devices and handsets. Our proposed approach, called *MultiScan,* is implemented entirely on the client-side. Unlike other prior work, MultiScan requires no changes in the Access Points (APs), nor does it require *a priori* knowledge of wireless network topology. Instead, MultiScan relies on using its (potentially idle) second wireless interface to opportunistically scan and pre-associate with alternate APs and eventually seamlessly handoff ongoing connections. In this paper, we first describe our implementation of MultiScan, and then present detailed evaluations of its effect on handoff latency and evaluate performance gains for MultiScan-enhanced wireless clients running Skype, a popular commercial VoIP application.

1. INTRODUCTION

IEEE 802.11 based wireless LAN (WLAN) technologies [1] are experiencing an unprecedented growth in the recent years fueled partly by decreasing costs and increasing data rates available through them. From the users' perspective, the key advantage of such access networks is untethered access — users can freely move within their area of coverage and stay connected.

In 802.11 WLANs, clients connect to the Internet via Access Points (APs). Due to various design choices and requirements of 802.11, as well as governmental regulations, the communication range of 802.11 devices is rather limited, and it is not uncommon for an AP (located indoors) to have effective communication range of less than 60 meters. Therefore WLAN coverage over a large geographic area is provided using multiple APs. Consequently, a wireless client moving through this area is likely to pass from the coverage area of one AP, to that of another. In order to maintain continuous connectivity during this period, the mobile client has to switch between APs, in a process known as a *handoff*. For mobile clients handoffs can occur very often.

The 802.11 standards does not completely specify handoff procedures. Depending the hardware and the vendor, it may take between 60 ms and 400 ms (262 ms on average) to complete a handoff and in some cases may result in connectivity gaps of up to a second [8, 12]. Such high handoff latencies are adequate for *discrete* client mobility scenarios where a client (typically a laptop user) uses the network when stationary, then moves to a different location but does not use the network during this move, and resumes network usage when stationary again. However, such handoff latencies are woefully inadequate for *continuous mobility* scenarios, where a client needs to use the network while mobile through the sequence of handoffs. Continuous mobility scenarios are of great significance in real-time latency-sensitive applications, e.g., Voice-over-IP (VoIP) and other synchronous multimedia applications. Poor handoff performance is one of the major hindrances to deployment of VoIP applications in WLAN scenarios.

The goal of this paper is to address this need of seamless mobility in WLANs thereby meeting the needs of VoIP and other such latency-sensitive applications. We propose a solution called MultiScan that uses two 802.11 network interfaces in the same device (an 802.11-based wireless phone or PDA). Our results demonstrate that such a mechanism is the only practical approach to *completely eliminate* handoff latencies in WLANs.

Why we need two radios?

We believe that a two radio interface solution is both practical and feasible and is the only mechanism that can eliminate handoff latencies in WLANs. While two separate radio interfaces in a single device may seem impractical (especially in a small form factor), it turns out that multiple commercial vendors are coming out with multi-band chipsets that allow communication or two or more channels, e.g., EN 301 intelligent wide-band WLAN chipset (see http://www.engim.com/). Hence we believe that two radio interface-based handoff solutions are both practical and timely, and can together jumpstart the process of efficient

*V. Brik and S. Banerjee are with the Department of Computer Sciences, University of Wisconsin-Madison, WI 53706, USA. Email: {vladimir,suman}@cs.wisc.edu. A. Mishra is with the Department of Computer Science, University of Maryland, College Park, MD 20742, USA. Email: arunesh@cs.umd.edu

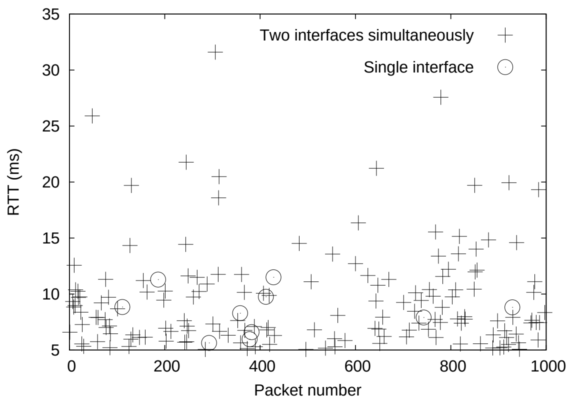

Figure 1: Reduced performance of a two-radio node when both radios used simultaneously versus only one radio is used at a time.

deployment of VoIP applications in WLANs.

Some prior research, e.g., Neighbor graphs [3] and SyncScan [2] have examined solutions to minimize handoff latencies for single-radio WLAN clients. In the Neighbor Graphs approach, extra functionality is implemented in both clients and APs through which APs infer the WLAN topology and use this information to reduce handoff overheads to around 30-40 ms. In the SyncScan approach, all clients and all APs in the network require appropriate time synchronization. While SyncScan can potentially reduce handoff latencies to a few milliseconds, their time synchronization requirements and other changes to the timing of AP "beacon" messages imply that they will not be applicable to legacy APs. More importantly such a solution cannot be used in various public community wireless networks (where neighboring APs are neither tightly-coupled nor centrally administered) that are emerging in different parts of the world today. Examples of such networks include Boingo Wireless, NoCatNet, Seattle Wireless, and Champaign Urbana Wireless Network in the US; Toronto TWCN in Canada; Manchester Wireless in UK; Sweden OpenNet in Stockholm, Sweden; Melbourne Wireless in Australia; and Miako.Net in Japan to name a few [1].

An important advantage of our proposed approach in MultiScan is that it requires changes only at the wireless client. In particular, we have currently implemented all the necessary functionality in a (Linux) client as a kernel module that controls the handoff process and (re)-association decisions. The MultiScan module is implemented over the driver interface of the wireless cards by using standard kernel API. By being independent of specific wireless cards, the MultiScan module is perfectly compatible with any wireless card (and its driver). We will make this module available for public use shortly. Finally MultiScan operates completely in the link layer and is invisible to applications.

In this paper we present the design of MultiScan, its prototype implementation, and demonstrate the usefulness of this approach for seamless handoffs in 802.11 WLANs. In particular we evaluate the performance of MultiScan through its performance of traffic floods as well as a popular commercial VoIP application, Skype (see http://skype.com).

[1] More at http://www.toaster.net/wireless/community.html

Exploiting multiple radios

Wireless nodes with multiple radios have more resources at their disposal than their single radio counterparts. However naive use of these additional resources can sometimes significantly hurt the performance at these nodes. This was demonstrated in some work by Adya et.al. [9] in the context of multi-radio mesh networks. In multi-radio mesh networks, wireless nodes are equipped with multiple radio interfaces and traffic takes multi-hop wireless paths through them. A possible use of two radio interfaces is to use them *simultaneously,* i.e., on every wireless hop, each node will use its two radios to form two wireless links with its neighboring node (also equipped with two radios) operating on different channels in the 802.11 ad-hoc mode with data traffic striped across these two links. In their work, Adya et.al. showed that when two such links between a pair of wireless nodes are used simultaneously TCP-based applications do not perform well due to re-ordering effects, unless the load of the two links are well balanced.

In our work we have found that using two radio interfaces simultaneously in a single device, especially those with small form factors, leads to significant loss of performance due to cross-interference between the radios. This is true even if the two interfaces of a 802.11 PDA or phone are operating on different 802.11 wireless channels and occur due to the physical proximity of transceiver circuitry of these interfaces. We demonstrate this in an experiment (see Figure 1) where we equipped a single node with two radio interfaces configured to non-interfering channels 1 and 11 respectively. Each of these interfaces were associated with a different AP. We ran two separate experiments — a two interface case, when both interfaces were involved in data transfer simultaneously, and a single interface case, where one of the interfaces was inactive. In each experiment, the active interface(s) performed a 'ping flood' to its corresponding AP — ICMP *Echo Request* packets were transmitted in sequence as soon as the *Echo Response* for the previous one came back (with packets assumed lost if no response was received 100 ms). The average ping latency on this link was about 1.6 to 1.8 ms. In Figure 1 we plot the tail of the round-trip time (RTT) distribution, i.e., the round-trip times for packets with RTT greater than 5 ms. It is instructive to see that the packets in the two simultaneous interface experiment experienced higher interference, as illustrated by the significantly higher number of packets experiencing high RTTs (about 154 packets for two simultaneous interfaces versus 10 for single interface).

Based on these observations, the design of MultiScan makes use of one radio interface as the primary data transfer interface, while the other (secondary) interface is used to facilitate a fast 'make-before-break' handoff as and when necessary, especially if the performance of the primary interface is considered to be deteriorating.

In the rest of this paper we will present specific design details on MultiScan and present detailed evaluation of this multi-radio approach as conducted in our wireless testbed. In particular we will look at how it helps eliminate handoff latencies and consequent improved performance for VoIP applications like Skype.

2. BACKGROUND

A typical WLAN consists of a number of APs. In order to reduce interference, neighboring APs operate on indepen-

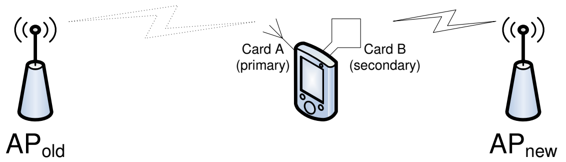

Figure 2: Multi-radio handoff scenario in MultiScan.

dent (non-interfering) channels. Different 802.11 standards have different number of such channels available, e.g., IEEE 802.11b has 3 such channels: 1, 6, 11. A client moving from the coverage area of one AP to another in a WLAN needs to change its association accordingly in order to stay connected. The ensuing handoff process consists of the following stages:

1. Scanning: In this stage the client monitors the wireless channels to identify available APs. APs periodically transmit *beacon* frames and clients locate APs on receiving such frames. Scanning can be either passive, where the client simply waits for beacons, or active, where the client actively solicits beacons. An AP can typically operate in one of many channels (e.g., 11 in 802.11b) and therefore a client in the scanning phase attempts to find APs in each such channel.

2. Authentication: Some organizations implement security mechanisms to restrict access to their wireless networks, e.g., using WPA standards. WLANs that implement such security mechanisms require clients to exchange appropriate authentication messages using the AP identified for association.

3. Association: In this stage the client associates with the new access point by sending an *Association Request* and receiving an association ID. If *Inter Access Point Protocol* (IAPP) [13] is implemented, the new AP will inform the old AP that the client is no longer associated with it and possibly obtain frames, buffered at the old AP, that are destined for this client.

A wireless interface engaged in the scanning process generally cannot be used for communication. In the scanning stage the interface has to switch from channel to channel to listen for AP beacons, and, it is likely that the new and the old APs are on different channels.

Prior work has shown that over 90% of the time in handoff process is spent in the scanning stage [6]. Because of this, work in optimizing handoff has focused on making scanning more efficient [6, 2]. In the next section we show that such optimizations are not critical (though still useful) to network nodes with multiple network interfaces, since the second interface can perform association-related tasks with a new AP.

3. HANDOFFS IN MULTISCAN

In the multi-radio scenario, we assume that a node has two interfaces, the *primary* interface and the *secondary* interface. Suppose that the primary interface is associated with AP_{old} and is used for communication, while the secondary interface is available to perform other tasks. Clearly, such multi-radio node will have an advantage since it will be able to communicate normally and perform management operations simultaneously.

In a naive approach, the secondary interface could perform the scanning stage (which is the most time consuming stage of a handoff), while the primary interface continues to communicate normally with its AP. Once the secondary interface determines an AP to which the node needs to connect next, the primary card could start the handoff process skipping the scanning stage. Using some recently proposed techniques this optimized handoff can be performed in less than 40 ms [8]. Besides the delay due to the last two stages of handoff, just switching the card to a different channel can require as much 20 ms [8], depending on chipset, which is significant for real-time applications. Although not the best we can do with multiple interfaces, this approach does vastly reduce latency due to handoffs. From the AP infrastructure's point of view, the node does not do anything unexpected, it simply appears as if it knows which AP to connect to without a scan.

In a more aggressive approach, we can virtually eliminate handoff latency if the secondary interfaces proceeds to associate with AP_{new} even while the primary interface continues to transfer data to and from AP_{old}. Once the secondary AP has finished its association, the roles of the two interfaces are swapped, i.e., the secondary interface starts functioning as the primary interface and the previously primary interface dissociates with AP_{old} and starts operating as the secondary interface. (see Fig 2). This is our approach:

1. Normal operation: Communication is performed using primary interface which is associated to AP_{old}, while the secondary interface is performing other tasks, including possibly scanning the channels.

2. Re-association: If it is determined that the it would be beneficial to switch to a new AP, the second card commences association with the new AP while the primary card is still used for data transfer with the old AP.

3. Interface Switch: As soon as the secondary interface is associated with the new AP, all of the node's the outgoing traffic is sent via the secondary interface. The primary interface effectively becomes invisible, but stays up for some time to receive packets that may still arrive through AP_{old} because they were buffered or due to slow bridging tables update.

4. Completion: Primary and secondary interfaces switch roles: the formerly secondary interface becomes primary and is used for communication, and the formerly primary interface is freed to be used for other tasks.

Clearly, such approach completely eliminates the handoff latency and is not disruptive to network connectivity. Nevertheless, just like in the vanilla handoff process, there is a possibility of dropped packets. This will happen to packets queued on the primary interface if the AP_{old} learns that the node is associated with a different AP will no longer accept packets from it. This can happen if the channel of the primary interface is congested.

Address management

An explicit goal of MultiScan is to require no changes in the APs or the wired infrastructure. In order to facilitate this goal, we require that both interfaces use the same IP address as well as the same MAC address. Standard utilities (e.g., iwconfig) allow clients to set MAC addresses of individual interfaces as desired. Therefore from the point of view of the infrastructure, it appears as if a single-radio wireless client just re-associated with a different AP (with zero latency). Note that when a handoff occurs, i.e., a secondary interface associates with a new AP and swaps functionality, the new AP automatically broadcasts a gratuitous-ARP in the

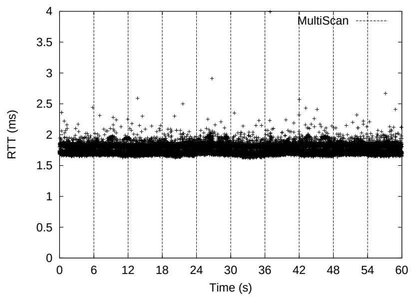

Figure 3: Ping RTTs with periodic handoffs for MultiScan client.

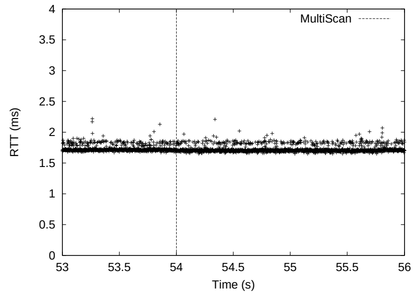

Figure 4: Zooming in on Figure 3 for a single handoff instant.

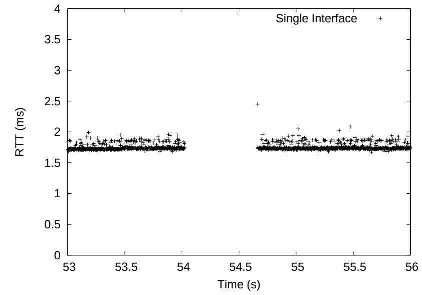

Figure 5: Zooming in on a single handoff in the ping flood experiment for single interface client.

LAN announcing the new association. This updates cached entries in different network devices.

4. EXPERIMENTAL EVALUATION

In our evaluation of MultiScan, we used a single laptop running (gentoo) Linux equipped with two Atheros 5212-based wireless cards as the client device. In our experiments, we set up two independent APs on the same host, using two independent wireless cards, each with an external antenna. The two antennae were widely separated so as it not cause any unintentional interference between each other (like in Figure 1). The two APs were set to operate on different 802.11b wireless channels, 1 and 11. The only reason for using the same host to function as two independent APs is to use the same clock for all measurements performed. Note that in other experiments we have configured different hosts (each with its own wireless card and external antenna) to act as APs and the performance of applications were no different in either case. There were no other 802.11 networks in the vicinity of the testbed.

Our MultiScan Linux module was installed in the client which controlled interface switching and roles of primary and secondary interfaces. MultiScan can trigger the interface switch operation in a flexible manner, e.g., when the signal strength of the first interface to its AP starts to weaken. In our experiments we performed both signal strength based switches as well as intentionally triggered switches — the latter allowed us more control in causing handoffs and we could stress test the performance of MultiScan using this feature. tcpdump traces were obtained at both the AP and the client and were used to measure necessary latency and other relevant parameters.

To test the performance of VoIP using MultiScan, we used a popular commercial software, Skype (version 1.0.0.20). In this section, in the interest of space we report on some of the interesting results from these experiments that illustrate the key performance aspects of MultiScan.

4.1 ICMP ping floods and handoffs

In the first set of experiments we use ping floods (that continuously sends *Echo Request* packets on the wireless link). The pings were sent from the wireless client to the AP(s), i.e., just across the wireless link and back. To stress test MultiScan, we performed an experiment in which 10 handoffs were performed by the MultiScan module within a one-minute period. Figure 3 illustrates the consequent RTTs of this stream of ping packets (usually between 1.6 to 1.8 ms). The vertical lines in the figure illustrate the instants the handoffs were initiated. From the figure it is apparent that the ping traffic experiences no perceptible latency *due to MultiScan handoffs.* A careful observer will notice that the degree of density of data points varies, depending on which card is used. This is not an artifact of MultiScan, but rather, our hardware (one interface is actually slightly slower than the other, either due to hardware or heat issues). To illustrate that the ping traffic experiences no perceptible handoff latency, we zoom into one representative handoff instant in Figure 4. We can easily compare this performance to that of a single interface based handoff experiment where we do not allow any modifications to the AP infrastructure [3, 2]. A zoomed in version of the single interface scenario is shown in Figure 5 which illustrates a 640 ms outage period (x-axis range is same as in Figure 4). The rate of traffic in ping floods is fairly high, especially when the ping traffic is across the wireless link only. Given the imperceptible change in performance for wireless handoffs when using MultiScan we feel confident that MultiScan will efficiently handle any traffic volume in the wireless link.

4.2 Skype and handoffs

We next present experimental results of running the Skype VoIP application on the same setting. The experiment was performed over a minute duration and 10 handoffs were triggered in this period.

In our experiments with Skype we have looked at two different metrics — end-to-end latencies of traffic and quality of audio as measured through the cross-correlation between the transmitted and received audio samples. We found that audio traffic imposed relatively low load on the wireless links — inter-packet latencies mostly varied between 10 and 30 ms. Given our results from ping flood experiments, we expected the Skype audio output at the receiver, transmitted using MultiScan, to be no different from the audio output at the source. In this paper we focus on the audio quality metric only (the latency metric look no different from ping flood experiments). In these experiments we used an audio signal corresponding to a person talking.

To quantify the differences in the audio we used to cross-correlation of the captured samples.

Cross-correlation of two real functions $f(t)$ and $g(t)$ is defined as: $f \star g = \int f(\tau)g(t+\tau)d\tau$. The cross correlation captures the similarity between the two functions. In particular two signals that are identical should have a high cross-correlation at the origin and no cross-correlation else-

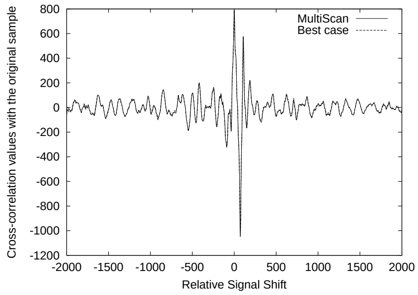

Figure 6: Comparison of cross-correlation in Skype audio application when using MultiScan with the best possible wireless performance.

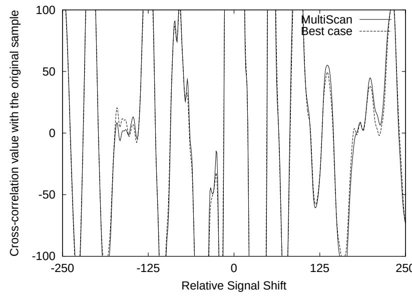

Figure 7: Zooming in on Figure 6.

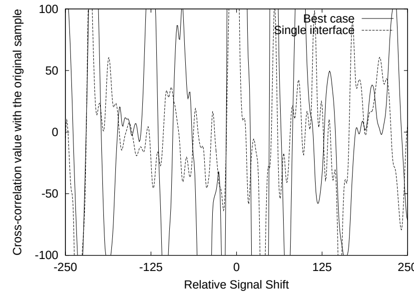

Figure 8: Comparison of cross-correlation in Skype audio application when single interface is used with best possible wireless performance.

where.

In each of Figures 6 to 8, each line represents the cross-correlation between the original audio signal (at the sender) and the received audio signal across the wireless link. (There are two such lines in each plot.)

The best case in all of these experiments reflect the situation when no handoffs are performed and the audio signal is transmitted uninterrupted over the wireless channel. This is the best performance achievable with Skype. Note that even in the best case, Skype performance is not perfect, just adequate for normal voice communication. In Figure 6 we show the cross-correlation function of two different scenarios — one in which there were 10 different handoffs over a minute interval and the client used MultiScan to perform the handoffs, and another was the best case just described. The two plots are virtually indistinguishable from each other. The maxima of the two cross-correlations are 788.3870 and 737.2810 for the best case and MultiScan respectively. There are actually very minor differences between the two plots especially given they were different runs on the wireless medium we zoom into the central part of the plot (shown in Figure 7). Again the differences are really too small to be perceptible and the audio quality sounds no different to the human ear.

To contrast this, we show the performance difference between the best case (single interface, no handoffs) to a case where we use a single card and the client has to perform just a single handoff (note that MultiScan performed 10 of them) in a minute's interval (Figure 8). We have zoomed into the same fragment of this cross-correlation plot as in Figure 7. We can observe that these two plots have significant differences, and the ensuing loss in audio quality is perceptible to the human ear.

Overall, we believe that these experiments demonstrate the efficacy of using MultiScan in eliminating handoff latencies and making it quite practical for deployment of VoIP applications in WLANs.

5. RELATED WORK

Researchers have used multiple radios to improve performance in a number of different applications. Some examples include, reducing energy consumption of wireless clients, e.g., Wake-on-Wireless [4], improving web performance in wide-area (cellular) networks, e.g.,work by Chakravorty et.al. [10], and in constructing wireless mesh networks, e.g., commercial ventures like MeshDynamics, and research efforts in Microsoft Research, Seattle [5] and Intel Research, Cambridge [11]. In particular Bahl et.al. [7] make an explicit case for multi-radio wireless systems for improved performance.

In this paper we take another step in advocating multi-radio wireless node design and demonstrate its applicability in WLANs to improve VoIP application performance. To the best of our knowledge, there has been no work in in eliminating handoffs in WLANs using multiple radios and demonstrating its use for VoIP applications.

Some other work in the past has focused on improving handoff performance using a single radio interface [3, 2]. Shin et.al. [3] in their Neighbor graphs work explore techniques to improve handoffs by implementing a topology inferencing technique in both clients and APs. Ramani et. al. [2] defined a technique called SyncScan that requires appropriate time synchronization between APs and clients. SyncScan also requires synchronization of *Beacon* broadcast times for different APs and periodic channel hopping of clients. The goal in both of these schemes is to reduce the channel scanning phase when a handoff occurs. Interestingly, through such changes and coordination between APs and clients, these schemes are able to reduce handoff latencies to about 40 ms and 2-3 ms respectively.

Unlike these above schemes that attempt to optimize performance with a single radio but requiring coordination and cooperation between neighboring APs, MultiScan to use multiple radios in wireless clients to completely eliminate handoff latencies. Our proposed schemes require no interaction or participation from APs, and hence can be deployed in arbitrary wireless environments, including environments where neighboring APs are not administered or controlled by a single entity. As discussed in Section 1, such scenarios are getting commonplace in many major cities around the world in form of community wireless networks. Table 1 summarizes the differences between these approaches and MultiScan.

Two other pieces of work from the recent past are also related to our efforts. Adya et.al. [9] defined a protocol called MUP, which allows multi-radio wireless nodes in a mesh network to potentially establish two separate wireless links between a pair of nodes. However, the authors advocate the use of only one of these links at a given time based on channel conditions. This work was primarily focused in

	Wireless interfaces	Handoff latency	AP infrastructure modification
Neighbor Graphs [3]	1	∼ 40 ms	yes
SyncScan [2]	1	2-3 ms	yes
MultiScan	2	0	no

Table 1: Comparison of different handoff mechanisms.

improving efficiency of wireless mesh networks.

Finally work by Chandra et. al. [14] demonstrated how a single wireless interface can be used to connect to two wireless networks simultaneously. Their approach is based on having the radio interface change channels unbeknownst to the applications. Their objectives and goals are orthogonal to that of ours.

6. CONCLUSION

It is not surprising that network nodes with multiple network interfaces can observe better network performance than nodes with a single network interface. Given many hard-to-overcome limitations of 802.11 wireless networking, such as short communication range, vulnerability to environmental noise, and relatively low throughput in many practical scenarios, this, coupled with the ever-increasing demand from the application side for bandwidth and low latency make it natural that such options be explored. While adding a radio interface leads to a modest increase in cost, our work demonstrates that significant performance improvements that are achieved. In particular we would like to reinforce the need for increased availability of multi-radio interfaces in wireless devices.

Overall, we make the following observations and contributions in this work:

- We recommend the use of two radio interfaces in eliminating handoff latencies in WLANs. Using two radio interfaces in wireless devices is already feasible but will be more so with the increased availability of multi-interface wireless cards.

- Our multi-radio approach does not use these interfaces in tandem for data transfer, as ensuing interference between the interfaces themselves (even when they are on independent wireless channels) can lead to degraded performance. Instead, one of the interfaces should be used as the primary data interface while the other serves as a secondary interface monitoring the environment for handoff opportunities. The functionality of the two interfaces are swapped when necessary.

- Utilization of multiple radios does not impose any additional load on wireless spectrum resources. This is because at any time one wireless interface acts as the secondary and does not impose any data load on the wireless medium. This also implies that the proposed mechanism is not hindered as more and more clients start operating in the multi-radio mode.

- We have developed MultiScan as a publicly available Linux module that will shortly be available for public downloads from:
http://www.cs.wisc.edu/∼suman/projects/multiscan

As a followup to this work, we intend to further experiment on how MultiScan should be extended to handle the newly defined Inter Access Point Protocol (IAPP) [13]. IAPP is a new mechanism proposed by the IEEE 802.11f working group to better handle roaming clients. (Currently IAPP is not widely implemented or available.) IAPP, among other requirements, enforces unique AP association and hence timing of AP switch operation currently implemented in MultiScan needs to be appropriately optimized.

7. REFERENCES

[1] IEEE, 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications 1999.

[2] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks" *Proceedings of the IEEE Infocom, March 2005.*

[3] M. Shin, A. Mishra and W.A. Arbaugh, "Improving the Latency of 802.11 Hand-offs using Neighbor Graphs" *Mobisys 2004 June, 2004, Boston, USA.*

[4] E. Shih, P. Bahl, and M. Sinclair "Wake on wireless:: an event driven energy saving strategy for battery operated devices" *ACM Mobicom, Oct 2002.*

[5] R. Draves, J. Padhye, and B. Zill "Routing in multi-radio, multi-hop wireless mesh networks" *ACM Mobicom, Oct 2004.*

[6] A. Mishra, M. Shin, and W. A. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process" *ACM Computer Communications Review, Apr. 2003*

[7] P. Bahl, A. Adya, J. Padhye, A. Walman "Reconsidering wireless systems with multiple radios" *ACM Sigcomm Computer Communications Review, Vol. 34, No. 5, Oct 2004.*

[8] A. Mishra, M. Shin, and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC layer Handoff Process" *ACM Computer Communications Review, vol. 33, no. 2, Apr. 2003.*

[9] A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou "A Multi-Radio Unification Protocol for IEEE 802.11Wireless Networks" *BROADNETS 2004, San Jose, CA, October 2004*

[10] R. Chakravorty, J. Chesterfield, P. Rodriguez, I. Pratt, and S. Banerjee "MAR: A Commuter Router Infrastructure for the Mobile Internet" *ACM Mobisys*, June 2004.

[11] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy "Experimenting with a Multi-Radio Mesh Networking Testbed" *Workshop on Wireless Network Measurements (Winmee), April, 2005.*

[12] F. K. Al-Bin-Ali, P. Boddupalli, and N. Davies, "An Inter-Access Point Handoff mechanism for Wireless Network Management: The Sabino System" *in Proceedings of the International Conference on Wireless Networks , Las Vegas, NV, June 2003*

[13] IEEE, 802.11f :IEEE Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation IEEE Standard 802.11f, 2003.

[14] R. Chandra, P. Bahl, and P. Bahl. MultiNet: "Connecting to Multiple IEEE 802.11 Networks Using a Single WirelessCard" IEEE Infocom 2004.