# 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 has been somewhat limited in wireless LANs today, partially because of the high handoff latencies experienced by mobile users. Our goal in this work is to eliminate handoff latency by exploiting the potential of multiple radios in WLAN devices. Our proposed approach, called *MultiScan,* is implemented entirely on the client-side, and, unlike prior work, MultiScan requires neither changing the Access Points (APs), nor having knowledge of wireless network topology. MultiScan nodes rely on using their (potentially idle) second wireless interface to opportunistically scan and pre-associate with alternate APs and eventually seamlessly handoff ongoing connections. In this paper we describe our implementation of MultiScan, 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 [1] based wireless LAN (WLAN) technologies have been 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 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 design choices and requirements of the 802.11 standard, the communication range of 802.11 devices is rather limited, and it is not uncommon for an AP to have an effective communication range of less than 60 meters indoors. Consequently, WLAN coverage over a large geographic area is provided using multiple APs, and a wireless client moving through such area is likely to pass from the coverage area of one AP to that of another. In order to maintain continuous connectivity, the mobile client has to switch between APs in a process known as a *handoff*. For mobile clients handoffs can occur very often because of the short range of APs.

The 802.11 standard does not completely specify the handoff procedure. Depending on the hardware and the vendor, it may take between 60 ms and 400 ms (about 260 ms on average) to complete a handoff, and in some cases a node may experience a connectivity gap of up to a second [7, 11]. Such high handoff latencies are adequate for *discrete mobility* scenarios where a client (typically a laptop user) uses the network while stationary, then moves to a different location but does not use the network during the move, and resumes network usage when stationary again. However, such handoff performance is highly inadequate for *continuous mobility* scenarios, where a client needs to use the network while mobile through a 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 WLANs.

The goal of this paper is to address the need for seamless mobility in WLANs, thereby meeting the needs of VoIP and other latency-sensitive applications. We propose a solution called MultiScan that uses two 802.11 network interfaces on the same device (e.g. an 802.11-based wireless phone or a PDA). Our experimental results demonstrate that MultiScan is capable of *completely eliminating* handoff latencies, and to our knowledge, it is the only existing practical approach that can eliminate handoff latency.

### Why do 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 physically 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 on two or more channels, e.g., EN-3001 intelligent wideband WLAN chipset for 802.11 networks (see http://www.engim.com/). Hence, we believe that approaches to handoffs that are based on using two radio interfaces are both practical and timely, and can jumpstart the process of efficient deployment of VoIP applications in WLANs.

Minimization of handoff latency for single-radio WLAN clients has been examined in prior research, e.g., Neighbor Graphs [3] and SyncScan [2]. In the Neighbor Graphs approach, extra functionality is implemented at both clients

---

*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
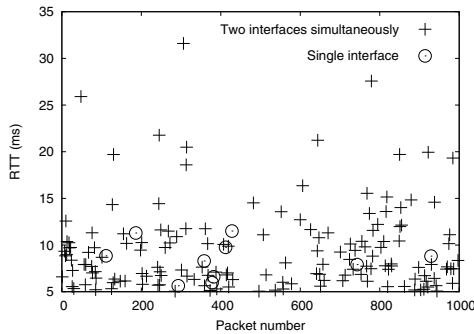
Figure 1: Performance of two radios used simultaneously.

and APs that is used to infer WLAN topology and reduce handoff overhead to around 30-40 ms. In the SyncScan approach, all clients and all APs in the network require time synchronization. While SyncScan potentially reduces the latency of a handoff to a few milliseconds, it requires regular suspension of communication for roughly twice the amount of time it takes a wireless interface to change channel, which, depending on hardware, could exceed 10 ms.

An important advantage of MultiScan is that it requires changes *only* at the wireless client. In particular, we have 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 relies only on the standard Linux kernel API and does not depend on device drivers. Therefore, MultiScan is completely compatible with any wireless card. Finally, MultiScan's operation takes place only in the link layer and hence is transparent to the applications. MultiScan module will be made publically available shortly.

In this paper we present the design of MultiScan, its prototype implementation, and demonstrate the usefulness of our approach for seamless handoffs in 802.11 WLANs. In particular, we evaluate the performance of MultiScan using traffic floods and a popular commercial VoIP application Skype [15].

## 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 could significantly hurt performance. This was demonstrated in a work by Adya et al. [8] in the context of multi-radio mesh networks, where 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 with data traffic striped across the two links. Adya et al. showed that when two such links

between a pair of nodes are used simultaneously, TCP-based applications perform poorly due to re-ordering effects, unless the loads 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 (such as a PDA or a handset), leads to significant loss of performance due to cross-interference between the radios. This is true even if the two interfaces 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.

We ran two experiments — a two interface case, where both interfaces were active, and a single interface case, where one of the interfaces was disabled. In each experiment, the active interface(s) performed a "ping flood", where ICMP *Echo Request* packets were transmitted about every 10 ms. The average ping round-trip time (RTT) was around 1.7 ms. In Figure 1 we plot the tail of the 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-interface simultaneous transfer experiment experienced higher interference, despite being on non-overlapping channels, as illustrated by the significantly higher number of pings with high RTTs (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, for example, if the performance of the primary interface is deteriorating.

In the rest of this paper we will present design details of MultiScan and a detailed evaluation of the multi-radio handoff approach, as conducted in our wireless testbed. In particular, we will look at how it helps to eliminate handoff latencies and thus improve 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 independent (non-interfering) channels. Different 802.11 standards have different number of such channels available, for example, in the US, IEEE 802.11b operates on 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 clients discover available APs by observing *beacon* frames that are periodically broadcast by APs. Scanning can be either passive, where a client simply waits for beacons, or active, where a client actively
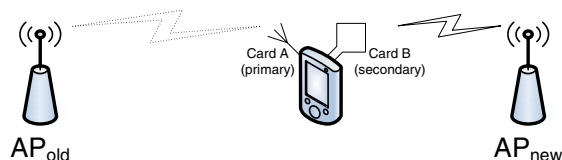
Figure 2: Multi-radio handoff scenario in MultiScan.

solicits beacons. An AP typically operates in one of many channels (e.g., 11 channels in 802.11b/g) and therefore a scanning client attempts to find APs in each channel.

*2. Authentication*: Access to 802.11 networks can be restricted by using such mechanisms as WPA. In these WLANs clients are required to exchange authentication messages with the AP identified for association.

*3. Association*: In this stage the client associates with the new AP by sending an *Association Request* and receiving an association ID. If *Inter Access Point Protocol* [12] is implemented, the new AP will inform the old AP that the client's association has changed, and obtain frames buffered at the old AP that are destined for the client.

In the scanning stage the interface has to switch between channels, and hence cannot be used for communication. Prior work has shown that over 90% of the time in the handoff process is spent in the scanning stage [7]. Because of this, work in optimizing handoffs has focused on making scanning more efficient [7, 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 be used to perform all 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 (see Figure 2). 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 is communicating normally with its AP. Once the secondary interface determines an AP to which the node needs to connect next, the primary interface could start the handoff process skipping the scanning stage. This optimized handoff can be performed in less than 5 ms. Besides the delay due to the last two stages of handoff, just switching the card to a different channel may require as much 20 ms [7], depending on chipset, which is significant for real-time applications. Although not the best we can do with multiple interfaces,

this naive approach vastly reduces latency due to handoff and is absolutely safe, since from the AP infrastructure's point of view, the node does not do anything unexpected, it simply appears as if the node knows which AP to connect to without a scan.

In a more aggressive approach, we can eliminate handoff latency if the secondary interface proceeds to associate with $AP_{new}$ while the primary interface is transfering data using $AP_{old}$. Once the secondary interface has finished its association process, 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. This is our approach:

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

*2. Re-association:* If it is determined that it would be beneficial to switch to a new AP, the second interface 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 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 arrive delayed from $AP_{old}$ because of buffering or a 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 potentially completely eliminates handoff latency (i.e. latency due to both the handoff process *and* switching the wireless interface to a different channel). Still, under certain conditions, connectivity during a MiltiScan handoff can be negatively impacted due to lost packets (though to a much lesser degree than during a single-interface vanilla handoff). Packets queued on the primary interface will be lost if $AP_{old}$ learns that the node is associated with a different AP and will no longer accept node's packets. This is can happen if the channel of the primary interface is much more congested than the channel of the secondary interface.

### 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 and MAC addresses. Standard utilities (e.g., ifconfig) allow clients to set MAC addresses of individual interfaces as desired. Therefore, from the point of view of the infrastructure, a MultiScan handoff appears as if a single-radio wireless client just re-associated with a different AP (with zero
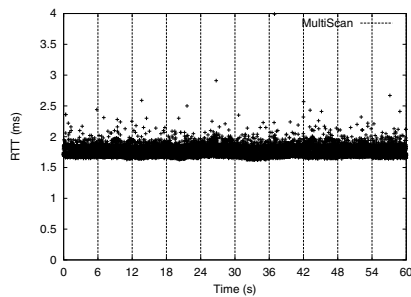
Figure 3: Ping RTTs with periodic handoffs for MultiScan client.
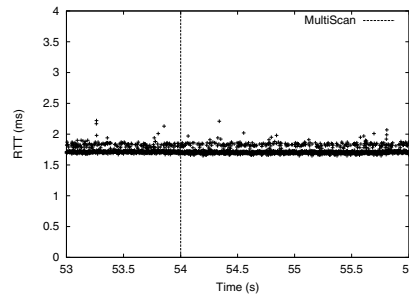
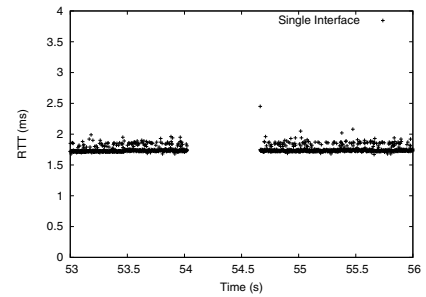Figure 4: Zooming in on Figure 3 for a single handoff instant.

Figure 5: A handoff by a single interface client (see Figure 4).

latency). Note that when a handoff occurs, i.e., a secondary interface associates with a new AP, that AP automatically broadcasts a gratuitous ARP in the LAN announcing the new association. This updates cached ARP entries in different network devices.

## 4 Experimental Evaluation

In our evaluation of MultiScan, we used computers running Gentoo Linux 2005.0, equipped with two Atheros AR5212-based wireless cards, operating on non-overlapping 802.11b channels (one on channel 1, another on 11). Antennae were separated so as not to cause interference as in Figure 1. Two independent APs were set up on one of the hosts (each one using its own wireless interface). A single computer hosted both APs so that the same clock was used for all measurements. Data gathered in experiments where each AP ran on a separate computer were no different than data presented here. There were no 802.11 networks in the vicinity of the testbed.

A MultiScan module installed at the client controlled handoffs and interface switching. MultiScan can trigger a handoff in a flexible manner, e.g., when the signal strength observed by the primary interface weakens. We performed both signal strength based handoffs as well as intentionally triggered handoffs — the latter gave us more control and allowed to stress-test the performance of MultiScan. tcpdump traces were obtained at both the APs and the client and were used to measure latency and other relevant parameters.

To test the performance of VoIP with MultiScan, we used a popular commercial software, Skype v1.0.0.20. In this section, in the interest of space we report only on some of the interesting results from our experiments that illustrate the key performance aspects of MultiScan.

### 4.1 ICMP ping floods and handoffs

In the first set of experiments we used ICMP ping flooding (in which *Echo Request* packets are continuously sent 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 experi-

ment in which 10 handoffs were performed by the Multi-Scan module in a one-minute period. Figure 3 illustrates the typical RTTs of a stream of ping packets. The vertical lines in the figure indicate the times when the handoffs were initiated. From the figure it is apparent that ping traffic experienced no perceptible latency increase *due to MultiScan handoffs*. A careful observer will notice that the degree of density of data points in the figure 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 experienced no perceptible handoff latency, we zoom in on one representative handoff instant in Figure 4. We next compare performance of MultiScan handoff to that of a typical vanilla single interface handoff in Figure 5. The figure shows 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 it goes 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

The experiments in this section consisted of transmitting a one-minute audio file of a person talking through Skype. We first established the baseline performance over a wireless link without handoffs, and then compared the results with data obtained with MultiScan and vanilla handoffs. We have considered two metrics: end-to-end latencies and audio quality. However, the latency data was no different than the data presented in the previous section. This is not surprising since Skype's natural inter-packet latency (15 to 30 ms) was higher than that of ping floods (about 10 ms). Therefore, here we will focus only on audio quality.

Informal qualitative analysis revealed that MultiScan handoffs had no perceivable impact on audio quality, while single-card vanilla handoffs resulted in long periods (about half a second) of dead air. To quantify the differences in the audio we used cross-correlation of the captured samples.

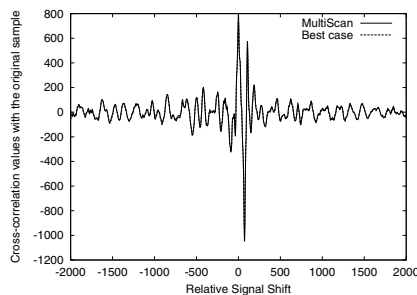Cross-correlation of two real functions $f(t)$ and $g(t)$ is

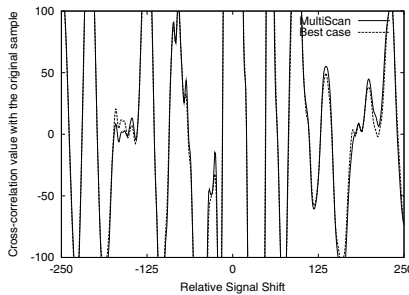Figure 6: Cross-correlations of Multi-Scan and best case scenario.

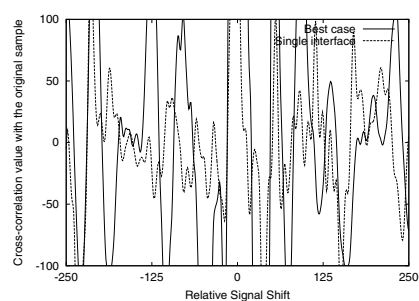Figure 7: MultiScan and best case (Figure 6 close-up).

Figure 8: Cross-correlations of single-interface handoff and best case scenario.

defined as: $f \star g = \int f(\tau)g(t+\tau)d\tau$. The cross correlation function captures the similarity between the two functions. In particular, two signals that are similar should have a high cross-correlation near the origin and low cross-correlation elsewhere.

In Figures 6 through 8, a line represents the cross-correlation between the original audio signal (at the sender) and the received audio signal across the wireless link.

The best case in all of the experiments reflects the situation where no handoffs are performed and the audio signal is transmitted uninterrupted over the wireless link. 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 MultiScan handoffs over a minute interval, and another is 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. These differences are very minor and are comparable to the differences between two independent best case (no handoffs) transmissions. Figure 7 is a close-up of Figure 6. The differences in cross-correlation values are small and the two samples sounded identical.

In contrast to Figure 7, consider Figure 8, where we show the performance difference between the best case (single interface, no handoffs) and a case where the client performed a *single* one-interface vanilla handoff. The two plots are significantly different and the loss in audio quality was perceivable to the human ear.

## 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 Rodriguez et al. [9], and in constructing wireless mesh networks, e.g., commercial ventures like MeshDynamics, and research efforts in Microsoft Research, Seattle [5] and Intel Research, Cambridge [10]. In particular, Bahl et al. [6]

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 eliminating handoff latency in WLANs using multiple radios and demonstrating advantages of such technique for VoIP applications.

Prior research has focused on improving handoff performance using a single radio interface. Shin et al. [3] in the 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. Both schemes seek to reduce the time spent in the channel scanning phase when a handoff occurs. By changing the APs and the clients, and by increasing coordination between them, Neighbor Graphs achieves handoff latency of about 40 ms, and SyncScan handoffs take 2-3 ms (but the technique requires periodic suspension of communication that could last more than 10 ms, depending on hardware).

Unlike the above schemes that attempt to optimize performance with a single radio but require coordination and cooperation between APs and clients, MultiScan relies only on multiple radios in wireless clients to completely eliminate handoff latencies. Our proposed scheme requires 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. Such scenarios are becoming commonplace in many major cities around the world in the form of community wireless networks [14]. Table 1 summarizes the differences between the approaches discussed above and MultiScan.

Two other works are related to our efforts insofar as they apply to multi-interface nodes in general. Adya et al. [8] defined a protocol called MUP, which allows multi-radio

| | Wireless interfaces | Handoff latency | Infrastructure modification |
|---|---|---|---|
| Neighbor Graphs | 1 | ∼ 40 ms | yes |
| SyncScan | 1 | 2-3* ms | yes |
| MultiScan | 2 | 0 ms | no |

* SyncScan requires routine suspension of communication that could last for more than 10 ms, depending on hardware.

Table 1: Comparison of different handoff mechanisms.

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 on improving efficiency of wireless mesh networks.

Finally, work by Chandra et al. [13] demonstrated how a node could stay connected to multiple wireless networks simultaneously. Their approach is based on having the radio interface change channels unbeknownst to the applications. In the context of MultiScan, this work applies to the potential functionality of the secondary interface.

## 6    Conclusion

It is not surprising that network nodes with multiple network interfaces can experience better performance than nodes with a single network interface. 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), coupled with the ever-increasing demand from the application side for bandwidth and low latency make it natural that multi-interface options be explored. While adding a radio interface leads to a modest increase in cost, this and other works demonstrate that significant performance improvements can be achieved. Therefore, 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 and will be more so with the increased availability of multi-interface and multi-band wireless cards.
- Our multi-radio approach does not use the radios 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 (periodically) monitoring the environment for handoff opportunities. The functionality of the two interfaces are swapped when necessary.
- Utilization of multiple radios does not create 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 clients start operating in the multi-radio mode.
- We have developed MultiScan as an open source 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 explore how MultiScan should be extended to handle the newly defined Inter Access Point Protocol (IAPP) [12]. 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.

## 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]  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.*

[7]  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.*

[8]  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*

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

[10]  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.*

[11]  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*

[12]  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.*

[13]  R. Chandra, P. Bahl, and P. Bahl. MultiNet: "Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card" *IEEE Infocom 2004.*

[14]  Public Wireless Community Network List
http://www.toaster.net/wireless/community.html

[15]  Skype Technologies, http://skype.com/