

Mobility Management Technique for Real Time Traffic in 802.11 Networks

¹Gurpal Singh, ¹Ajay Pal Singh Atwal and ²B.S. Sohis
¹Department of CSE and IT, BBSBEC, Fatehgarh Sahib, India
²Director, UIET, Panjab University Chandigarh, India

Abstract: A practical mobility management technique aimed at reducing handoff latency to less than 10ms has been proposed and implemented in this paper. Handoff latency has been reduced by decoupling scanning from handoff execution process. This scheme has been implemented in the form of client side application with modifications to open source Madwifi new generation driver. Performance of this mobility management scheme has been evaluated on experimental Testbed for finding its suitability for real time applications. Effect of the proposed scheme on commercial VoIP client, Skype has been evaluated. Effect of the proposed handoff and optimized background scan on audio and video streaming has also been investigated. The results indicate that this mobility management technique can be used for real time traffic over 802.11 wireless networks.

Key words: VoIP, Darwin Streaming Server, Handoff, Background Scanning

INTRODUCTION

In recent years there has been an explosive growth in the use of wireless local area networks arising from the advent of the IEEE 802.11b standards^[1]. Service providers have nowadays started to deploy real time voice and Streaming multimedia services over 802.11 wireless networks. Bandwidth issues in the wireless networks has taken backseat as emerging standards such as 802.11n can provide data rate to the range of 500Mbps. But, limited range of 802.11 access points lead to frequent handoffs in case the user is mobile. Availability of small portable devices with wireless interfaces has increased the suitability of 802.11 wireless networks for providing real time services to the users on the move. However available 802.11 handoff mechanisms are still far from stringent quality of service demands imposed by the real time services. This work has proposed and implemented a practical mobility management technique to reduce handoff delay, jitter and packet loss to a level well accepted by real time voice and multimedia streaming services. Scanning and neighbour discovery phases have been separated from the handoff execution. Effect of scanning and neighbour discovery phase on voice and multimedia streaming traffic has also been evaluated. Effect of this handoff scheme on VoIP^[21] sessions and on audio and video streams have been experimentally evaluated. Commercial VoIP client Skype and Darwin streaming server^[12] have been used testing suitability of the proposed scheme for voice and

audio/video streaming applications. All the experiments have been performed on experimental Testbed set for the purpose.

HANDOFF PROCESS IN 802.11

IEEE 802.11 specifies physical and medium access standards for wireless local area networks and leaves the handoff and roaming issues to vendor implementation. Vendor implementations are normally proprietary in nature. In this paper, generic handoff procedure followed by the 802.11 wireless networks has been presented. At handoff times, mobile nodes after losing connectivity with the current access point, goes into scanning phase and scan all the available 11 channels for discovering the access points working in the neighborhood. In conventional 802.11 handoff, neighborhood discovery and handoff process are tightly coupled and is responsible for long handoff latency at handoff times. Various phases of the conventional 802.11 handoff process include scanning, authentication and reassociation. Scanning^[2] phase is the most time consuming phase and a major contributor towards 802.11 handoff latency. In scanning phase mobile node tunes its radio interface on all the available 802.11b/g channels one by one and tries to extract the signal strength and other useful handoff parameters for those channel. Scanning techniques are broadly classified into two categories namely, active and passive scanning^[11]. In passive scanning a mobile node switches to a particular channel and wait for the beacon frames to arrive on that channel. Beacon frames are

normally sent by access points after every 100ms interval. In active mode of scanning, a mobile node after switching to a particular channel sends a probe request on that channel. Probe request is immediately answered by the access point operating on that channel (if any) in the form of a probe response frame. Active scanning though fast generates signaling traffic on the wireless channel. Scanning phase takes around 350-1000ms.

Authentication is a process through which the mobile node tries to prove its identity to the access points. This is done by sending authentication frame to access point, which can accept or reject the request based on some policy. Time to authenticate is normally less than 10ms. This time can be further reduced by using pre authentication techniques suggested by [7]. Association is followed by successful authentication and involves sending re-association request frames by the mobile node to access point. The access points respond by sending a re-association reply. After successful association, a mobile node is assigned proper association identity and required resources by the concerned access point. Association process normally takes less than 10 ms of time.

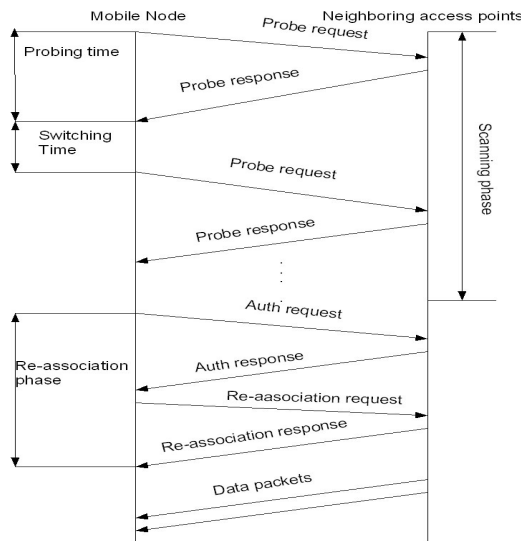


Fig. 1: Representation of scanning, authentication and re-association in 802.11 handoff process

Fig. 1 shows the timing chart of conventional 802.11 handoff process showing the scanning, authentication and re-association phases. It can be easily concluded that scanning is major contributor to 802.11 handoff delays and as such required to be decreased for providing seamless mobility. Therefore, most of the existing techniques for reducing the handoff delay in

802.11 wireless networks focus primarily on decreasing the time required to complete the scanning phase.

Latency for conventional handoff varies anything from 300-500ms and is far from being suitable for running voice and multimedia services over wireless networks. Major impediment in reducing the handoff latency is the long scanning phase. Thus for reducing 802.11 handoff latency, scanning phase has to be decoupled from the handoff execution. In this paper we have broken the long scanning phase at handoff time into smaller phases and interspersed it in the normal data communication. In this scheme, mobile nodes use Background Scan mechanism for neighborhood discovery and for ascertaining signal strength of the access points operating in the neighborhood, without losing connectivity with the current access point.

Neighborhood discovery and scanning phase at handoff time is major contributor towards the long handoff delays in IEEE 802.11 Networks. Such long handoff delays result in packet loss, delays, jitter, retransmissions and other unwanted affects. To make such networks suitable for real time applications, handoff delay should be reduced to less than 10ms. Prior discovery of access points operating in the neighborhood can be highly useful in reducing the handoff delays. Numerous attempts have already been made to gather prior neighbour information and using the same for layer 2 handoff delay reduction. But most of these approaches require either modification to network infrastructure or firmware upgrades. A brief overview of the work done by various researchers in this area has been summarized below.

In [4], Ishwar Ramani and Steven Sage have proposed a practical neighbour discovery and handoff approach where all the mobile nodes and access points are assumed to be synchronized with respect to time using Network Time Protocol (NTP). Access points working on different channels send beacon frames staggered in time. Mobile node tries to discover information about the neighborhood by switching channels precisely at times when beacons from neighbouring access points are expected. This approach is practical and works without any modification to network infrastructure but requires stringent time synchronization which is difficult to achieve. Yong Liao and Lixin Gao [5] have proposed a smooth handoff algorithm to discover neighbours by breaking the long discovery and scanning phase into smaller sub phases. In this approach, instead of scanning all the 11 channels at one go, channels are divided into smaller groups and single group of channels is scanned at a time. This scheme uses two signal threshold levels, one for handoff and second for initiating the discovery phase. But, authors

have not studied the effect of discovery phase on ongoing traffic and moreover, setting the threshold level for triggering the discovery phase would be a black art. In another approach [6] Sajal K. Das et al have proposed and implemented a service discovery and fast handoff scheme by modifying Madwifi drivers (old generation) on Atheros chipset based cards. This approach provides low handoff latency and jitter and does not modify the network infrastructure but works with old generation Madwifi drivers only. In [7], authors have proposed a neighbour graph approach where each access point gathers information about its neighbouring access points. Later, this information is disseminated to the mobile nodes and only neighborhood channels need to be scanned. A firmware upgrade to all the access points is required to construct and maintain such neighbour graph, which makes this approach less interesting and difficult to deploy on large number of existing 802.11b/g access points. In [8] authors have developed smart algorithms to trigger handoff in 802.11 systems and used the leakage power to capture information of APs located in the overlapping frequency. IEEE is currently working on 802.11k [9] standard which plans to introduce the concept of Neighbour Reports. It will take some time before we could expect the standard to come into being. Complete implementation of 802.11k would require a significant investment and infrastructure changes for a large volume of existing wireless networks. In [10], the authors' claim of achieving zero seconds handoff latency using two wireless cards so that one card can be used for Background Scanning while the other card can be used for packet transmission. But, implementing this solution involves additional hardware costs and more drain on power, which is a scarce resource in mobile environments. Moreover, very little efforts have been made to study the effect of handoff on multimedia real time traffic in 802.11 networks. Aim of the present work is to develop and implement a client side application for managing neighbour discovery for achieving fast handoff in 802.11 mobile nodes without making any modifications to either network infrastructure or IEEE 802.11 standards, and to experimentally evaluate its performance on real time voice and multimedia traffic.

Background Scan is a process through which the mobile node performs neighborhood discovery and measure signal strength of neighbouring channels without breaking connectivity with the current channel. In background scan phase, a mobile node sends power save (PSM) frame to the current access point and meanwhile shifts to neighbouring channels to perform scanning. Once scanning on neighbouring channel is complete, the mobile node returns to the current channel. Current access point will forward any buffered

traffic it has received, when the mobile node was in PSM mode. But, the default Madwifi driver is not capable of performing background scan in presence of traffic and is started only if there is no traffic for the mobile node for last 250ms (default value). Also background scan is cancelled in case traffic resumes over the link. We have made modifications to the default Madwifi driver [20] to make background scan work even in presence of traffic. Further, to reduce the number of delayed packets, Background Scan operation is performed on a single channel at a time. Figure: 2 shows an example of background scan operation being performed on a single neighbouring channel. Our experiments shows that Atheros 5212 [19] chipset takes around 4ms of time for channel switching and mobile node takes additional 8ms to receive the probe response, even in the presence of heavy traffic on the neighbouring channel. Thus the wait time has been fixed to 8ms. Therefore as shown in Fig: 4, total time required to complete background Scan operation on a single channel takes around 16ms.

Probability of packet being buffered on access point during background scan operation depends on the time to complete background scan (BGSCANT) and packet inter arrival time (IAT) as shown below.

$$P(\text{packetbuffered}) = \begin{cases} \frac{BGSCANT}{IAT} & \text{if } BGSCANT \leq IAT \\ 1 & \text{Otherwise} \end{cases}$$

Where $BGSCANT = 2 \cdot \text{ChannelSwitch Time} + \text{WaitTime}$

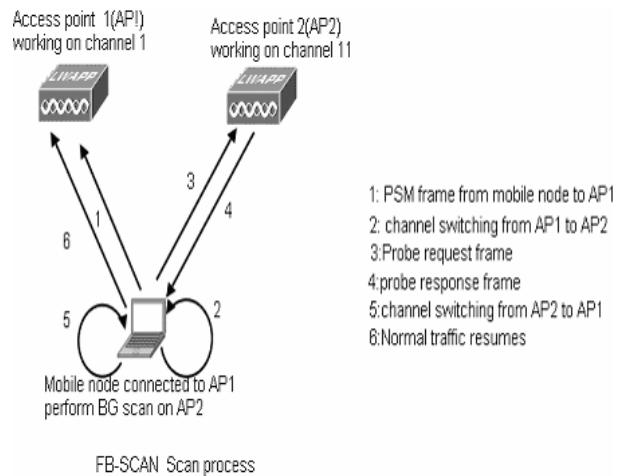


Fig. 2: Representation of Background Scan operation for a single neighbor

Measurements for Atheros 5212 chipset based WG 511T Netgear PCMCIA card shows channel switching time to be around 4ms. Based on previous studies [2, 11]

and our experimentation, wait time of 8ms is sufficient for probe response to arrive on channel. Keeping this in view background scan operation on a single channel requires at most 16ms of time to complete. Any packet that arrives within this time will be buffered at the access point. Probability of packet being buffered at access point depends on three factors namely; time to complete background scan operation, packet IAT and time at which background operation is triggered. In this implementation, background scan operation is synchronized with packet arrival or packet departure depending on whether station is transmitting data or receiving data. By doing so, the possibility of background operation being completed before the arrival of next packet is increased. Hence, the probability of packet being buffered and consequently being delayed is decreased.

PROPOSED MOBILITY MANAGEMENT

Basic purpose behind development of this mobility management technique is to minimize the impact of handoff and associated neighbour discovery operations on ongoing real time voice and multimedia applications. In this paper, default background scan process of Madwifi driver has been modified and used for discovering and updating the information about the neighbouring access points and their channels. Once the neighbour information is available with the mobile node, handoff process gets reduced to mere switching over to one of the most suitable access point and thereby scanning is totally eliminated at the handoff times. Purpose of background scan is to decouple scanning from the handoff. In background scan algorithm, scanning is performed on neighbouring channels after discrete intervals of time while the mobile node is still connected to the current access point. This eliminates the need for scanning at the handoff time and consequent reduction in handoff delay and packet loss. Working of this scheme is explained in the flow chart shown in Fig: 4. Mobile node after shifting to new access point, loads all the 11 channels to the background scan channel list and starts performing background scan operation by selecting a single channel at a time. Background Scan operations are performed after fixed interval of time and iterated over all the channels present in the background scan list. Experimental measurements show that Atheros 5212 chipset takes around 4ms (driver log messages) of time for channel switching. Mobile node immediately sends Probe request frame after switching to one of the channels present in the Background Scan list. Access points respond by sending a probe response frame. Our

measurements show that 8 ms of time is enough for the mobile node to wait for Probe response frame. If no probe response frame is received on a channel during wait time, it is removed from Background Scan list and next channel is selected for Background Scan operation. After completing a Background Scan on a given channel, the node takes another 4ms to switch back to the original channel. Thus total time (2 channel switch operations and wait time) required to complete a Background Scan operation on a single channel is around 16ms. It has been found that commercial VoIP clients generate traffic at rate no more than 20 ms IAT. Thus, even in the worst case, background scan operation on a single channel could be safely completed within packet inter arrival time (IAT) between the two consecutive packets, provided background scan operation is started in synchronization with the packet arrival or departure. Further to prevent any packet loss, Power save mode (PSM) frame is sent on current channel so that any packets destined for the mobile node during Background Scan time are buffered at the access point. After the first iteration, the background scan channel list is reduced to number of channels actually heard (probe response frames received) by the mobile node and future background scan operations are performed on these channels only. To prevent missing of any beacon frame on the current channel; background scan is never conducted when beacon frame on the current channel is expected. Background Scan process can thus be used for discovering and continuous updating of the neighborhood information in 802.11 wireless networks. The continuously updated neighborhood information is stored in the application cache and is fed to handoff application for making handoff decisions.

This implementation has the following key features and advantages.

1. Background Scan and handoff process is entirely controlled by a single client application written for the purpose and installed on the mobile node.
2. Information about the neighbouring access points and their channels is discovered using background scan operations and stored in application cache.
3. Background scan neighbour discovery approach works successfully even in presence of traffic
4. To prevent any packet loss, background scan operation is synchronized with packet reception or transmission depending on whether node is transmitting or receiving so

that background scan operation is completed before arrival of next packet.

- It is backward compatible and implemented without any modifications to network infrastructure or IEEE 802.11b/g Standards.

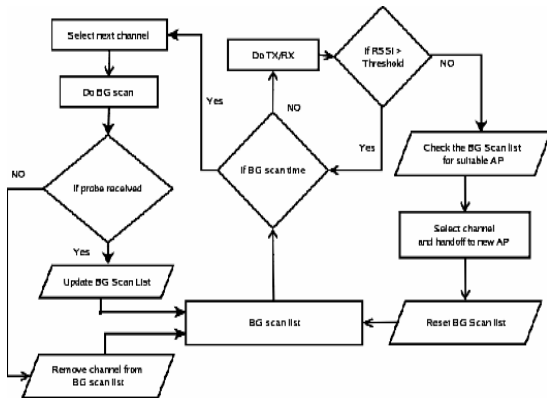


Fig. 3: Flow chart representing the working of the proposed Background Scan approach

HANDOFF EXECUTION

Handoff process in 802.11 is normally controlled by the driver and is executed when connectivity with the current access is broken due to poor signal reception. Thereafter, the mobile node goes into the scan phase to find out the access points operating in the neighborhood. But in this case, since the neighbour information and signal strength has already been made available to the mobile node; handoff decision is taken by the client application before the connectivity from the access point breaks. Handoff and neighbour discovery in this approach is entirely controlled and decided by the client application. Client application performs background scan on all the channels in the application neighbour cache list on one channel at a time basis and the resultant Received Signal Strength Indicator (RSSI) values are stored in the application cache. Time interval between two successive background scan operations can be varied depending on user mobility and variations in RSSI values. RSSI value of the current access point channel is also monitored continuously and its weighted moving average is calculated by the client application. Handoff execution is performed once the RSSI value of the current access point reaches the threshold value (pre assigned value). Handoff is executed by switching to access point channel offering the highest RSSI value from the

application cache and by subsequently sending authentication and association requests. Therefore, this approach completely decouples the scanning phase from handoff process. No scanning is performed at handoff time and thus the long scanning phase is entirely eliminated.

IMPLEMENTATION AND EVALUATION

Detailed experimental investigations have been conducted to study the effect of client application controlled handoff and background scanning on background voice and multimedia traffic with special reference to handoff delay, packet loss and packet retransmission. Impact of background scan on real time traffic has been investigated using commercial VoIP client and audio and video streaming. Description of experimental Testbed shown in Figure: 4 has been summarized below.

- Linksys WRT54GL and Dlink-2100AP with Dlink DES-1016D switch.
- GNU/Linux (kernel 2.6.18) based laptops equipped with Atheros 5212 chipset based Netgear WAG 511 PCMCIA wireless cards, using the modified Madwifi new generation drivers (release 1886-20070107), as mobile nodes [15, 18, 19, 20].
- Darwin streaming Server [12] and Skype [21] for traffic generation.
- Wireshark Network Protocol Analyzer (0.99.3a)^[17] for traffic capture.
- Our Custom applications for analysis of captured traffic using libpcap and libpcapnav^[22].

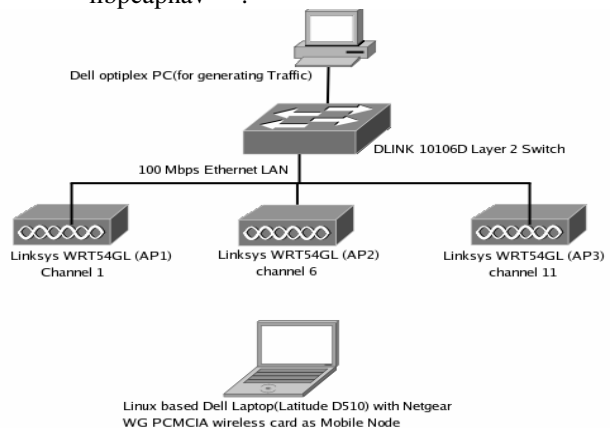


Fig. 4: Representation of experimental Testbed

Mobile node is initially attached to access point working on channel 6 and it perform handoffs by switching to neighbouring channels 1 and 11 as shown in the Fig : 4. Similarly Mobile node can do neighbour discovery by performing background scan on channel 1 and channel 11. Experimental results for the following scenarios have been obtained and compared:

1. Effect of Handoff and Background Scan on commercial VoIP Traffic
2. Effect of Handoff and Background Scan on audio streaming
3. Effect of Handoff and Background Scan on video streaming.

Effect on VoIP Traffic: For the first scenario, VoIP session is generated using popularly used commercial client, Skype [21]. VoIP session is started between the mobile node and desktop machine in the wired network. Handoff is executed while the Skype session is underway and results have been obtained in the terms of packet delay and packet loss at handoff time. Wireshark network protocol analyzer has been used to capture the Skype traffic and we have used custom application for analysis of the captured packets. The results obtained thereof are represented in form of packet CDF for normal VoIP traffic and VoIP traffic in presence of handoff in Fig: 5. we have collected packet trace of around 15ms duration with three handoffs. The results show that due to handoff, there is slight increase in the number of packets arriving either delayed or advanced from the packet mean inter arrival time. These delayed packets can be attributed to the handoff times. Also as per our analysis no packet loss has been observed at handoff times.

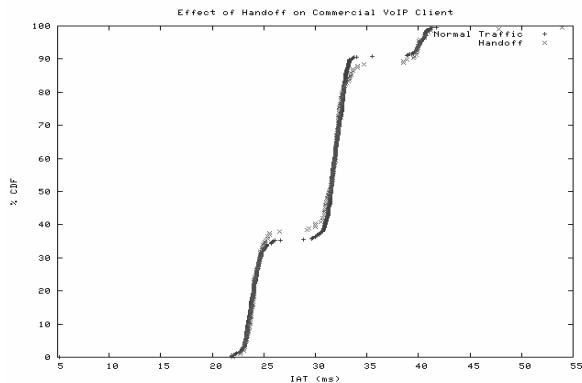


Fig. 5: Effect of Handoff on Commercial VoIP client Skype

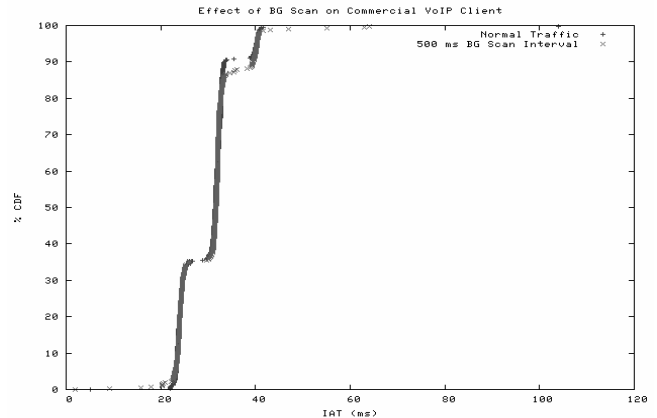


Fig. 6: Effect of background scan on Skype VoIP session

Our approach uses background scan for neighbour discovery and as explained earlier Background Scan leads to disruption in communication with the current access point. Therefore, we have also investigated the effect of background scan on the ongoing skype session. In this case, we have compared the CDF of normal Skype traffic with Skype traffic in presence of background scanning. Results as shown in Fig: 6 indicate that in presence of background scan, slightly higher percentage (around 4%) of packets is delayed from the mean IAT as compared to the normal traffic. This is due to buffering of the packets at access point during background scan operations. This effect is very minimal and leads to conclusion that our handoff and Background Scanning schemes are VoIP ready

Effect of Handoff on Audio Streaming: Audio streaming is another important service which can be deployed over wireless networks in the form of song over demand, web casting, and internet radio. Keeping this in view, investigations on the suitability of the proposed handoff and background scanning mechanism for audio and video streaming have been conducted. For generating audio stream, Darwin streaming server has been used. Darwin Streaming Server (DSS), the open source version of Apple's QuickTime Streaming Server [12,13], allows sending streaming media to clients across the Internet using the industry standard Real time protocol (RTP) and Real time streaming protocol (RTSP) protocols. It can stream MPEG-4, 3GPP audio and video and the source for DSS is available under the Apple Public Source License. The media packets are sent in real time, so that a one-minute movie is sent over the network in one minute. The packets are time-stamped, so they can be displayed in time-synchronized order. RTP streaming works with

live content in addition to previously-recorded movies. It can also carry a mixture of the two. Real-time streams can be sent one-to-one (unicast) or one-to-many (multicast) using DSS. Audio stream generated by the streaming server to media player client on the mobile node is captured using Wireshark protocol analyzer. Audio packets captured thus are analyzed using our custom application. In first case, the mobile node undergoes handoffs in presence of background audio streaming traffic. Figure: 7 and Fig: 8 show the packet IAT time for normal audio stream and audio stream with handoff operations. Results indicate that handoff doesn't cast any adverse effect on audio stream. In case of background scanning around 5% more packets are delayed in presence of background scanning as compared to the normal audio stream as shown in Fig: 10. This is due to buffering of packets at the access point during background scan operations. Figure: 9 show the overlapped packet CDF versus time for normal audio streaming and audio stream in presence of background traffic.

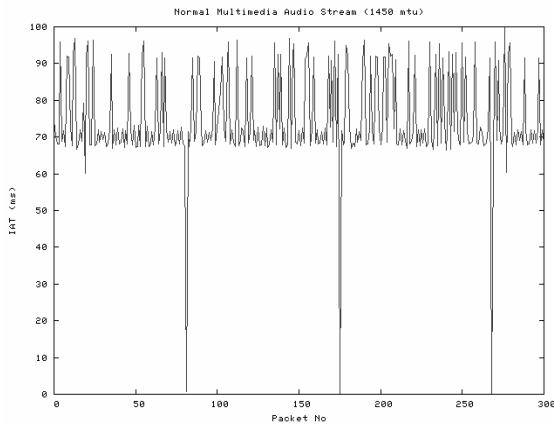


Fig. 7: Packet IAT for normal Audio stream

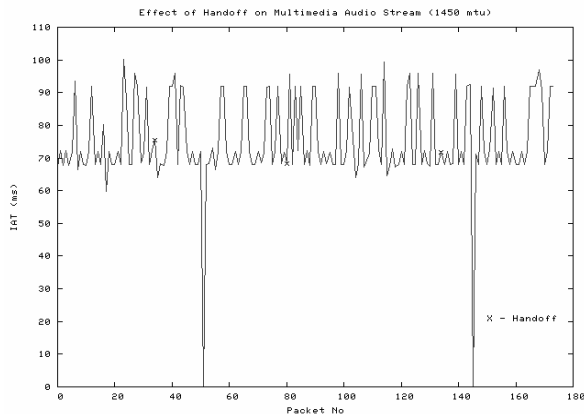


Fig. 8: Packet IAT for Audio stream in presence of Handoff

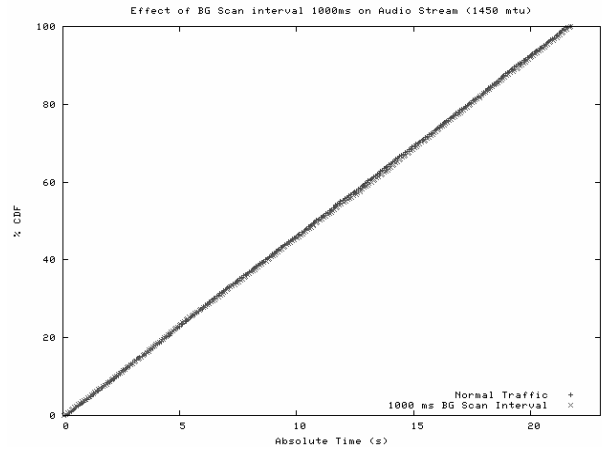


Fig. 9: Packet IAT versus time for Audio stream in presence of Handoff

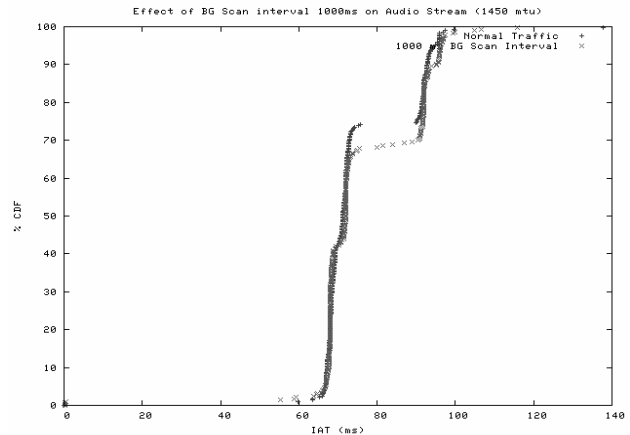


Fig. 10: Packet CDF for audio stream (normal and in presence of background scan)

Effect on video streaming: Video streaming is a client server technology which provides for multimedia data to be transmitted and consumed. The main goal of the streaming is that the stream should arrive and play out continuously without interruption. However, this requirement is constrained by fluctuations in the network. Handoff and background scanning times also lead to such fluctuations and hence their effect needs to be studied on the mp4 video streams. In this experiment, open source Darwin streaming server has been used to generate mp4 video stream to mobile node [14]. For streaming mp4 file a hint track has to be prepared that indicates the server how the contents should be streamed. Hint tracks [16] basically allow a server to stream media files without requiring the server

to understand the media types, codecs or packings. We have used the maximum transfer unit (MTU) size to 1450 bytes in this experiment. Wireshark network protocol analyzer has been used to capture the traffic on the mobile node. Captured packets are analyzed using our custom application

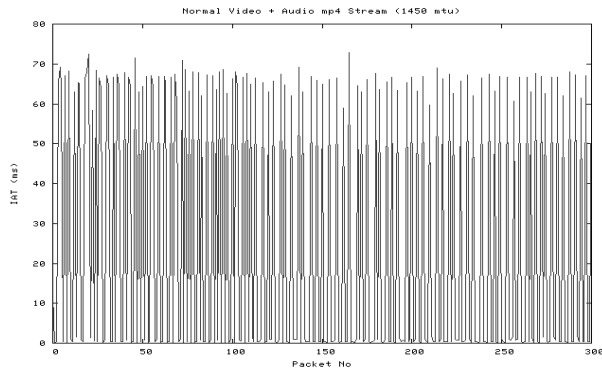


Fig. 11: Frame IAT for normal Audio mp4 stream

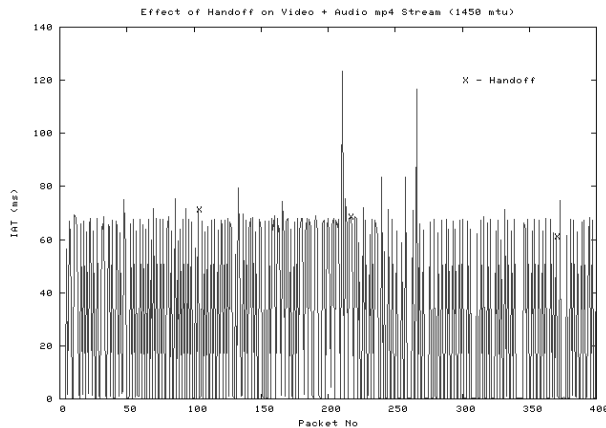


Fig. 12: Packet IAT in presence of three handoffs in presence of audio mp4 stream

Results have been obtained in the form of packet IAT and cumulative CDF for normal traffic and traffic in the presence of handoff and Background Scanning. Figure: 11 shows the normal packet IAT versus packet number for the normal mp4 steam captured at the mobile node. Fig: 12 show three handoff operations being performed in presence of background mp4 video stream. Results show that handoff operation cast very minimal effect on the packet IAT time.

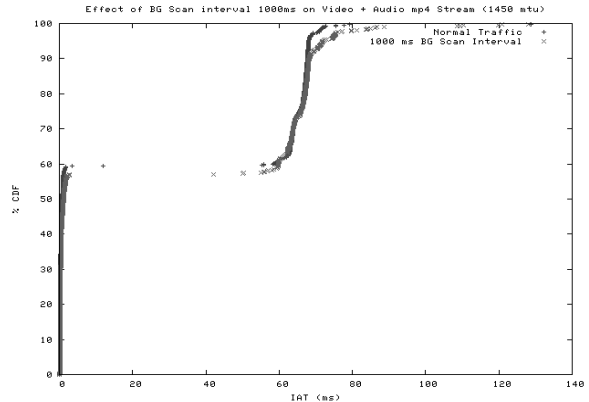


Fig. 13: Packet CDF for mp4 audio and video stream (normal and in presence of Background Scan))

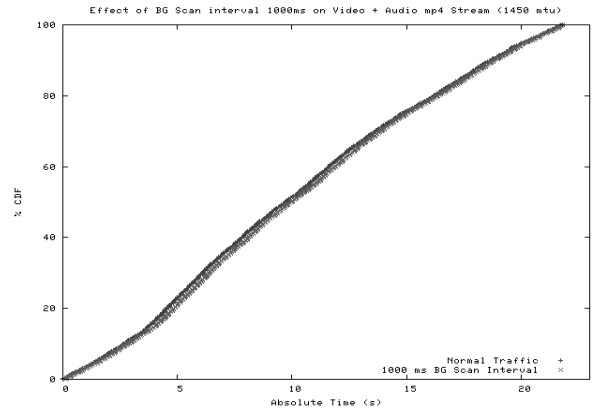


Fig. 14: Packet CDF versus time for Audio+Video stream (normal and with Background Scan)

Background scanning as discussed earlier leads to temporary disruption of communication with the current access point. Therefore the packets that arrive during the background scan operation times are buffered at the access points and delivered to mobile node when it come back on the current channel. This buffering of the packets leads to packet being delayed. This effect has been shown in Fig; 13 in the form of overlapped packet CDF for normal mp4 stream and mp4 packet stream in presence of background scanning. Packet CDF for normal video stream shows that around 60% percent packets are arriving immediately and rest 40% are coming around 60ms. In case of background scanning the percentage of delayed packets is very slightly increased. This shows that background scan cast very minimal effect on ongoing .mp4 video stream. Fig: 14 show CDF of packet versus time for normal .mp4 stream and that in presence of background. Both the lines are almost overlapping

CONCLUSION

Voice and multimedia applications are highly intolerant to the delayed and the lost packets. This work has proposed and implemented a multimedia ready handoff and neighbour discovery mechanism for 802.11 wireless networks. Implementation has been made available in the form of client application which can be incrementally deployed without any modifications to the network infrastructure. Results indicate that handoff is completed in less than 10ms of time and background scanning on a single channel takes around 16ms. Experimental study of effect of the proposed approach on voice and multimedia traffic shows that the scheme is multimedia ready. Moreover, this approach does no modifications to 802.11 standards and has been implemented using off the shelf hardware and open source software. Also the proposed neighbour discovery scheme can supplement any of the existing handoff algorithms in making fast and lossless handoff decisions.

REFERENCES

1. IEEE Computer Society LAN MAN standards Committee, "IEEE Standard for Information Technology: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
2. 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.
3. S. Pack and Y. Choi, "Fast inter-ap handoff using predictive authentication scheme in a public wireless LAN. In *Proceedings of IEEE Networks Conference (conjunction of IEEE ICN and IEEE ICWLHN)*, Atlanta, GA, Aug. 2002.
4. I. Ramani, and S. Savage, "Syncscan: Practical Fast Handoff for 802.11 Infrastructure Networks," in *Proceedings of the IEEE Infocom Conference 2005, Miami, FL, USA*, March 2005.
5. Yong Liao, and Lixin Gao, "Practical Schemes for Smooth MAC Layer Handoff in IEEE 802.11 Wireless Networks," in *proceeding of IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006)*, Buffalo, NY, 26-29 June, 2006.
6. <http://crewman.uta.edu/corenetworking/projects/handoff/nwbandoff.html>.
7. M. Shin, A. Mishra, W. A. Arbaugh, "Improving the Latency of 802.11 Hand-offs using Neighbor Graphs," in *Proceedings of the ACM MobiSys Conference*, Boston, MA, USA, June 2004.
8. V. Mhatre and K. Papagiannaki, "Using Smart Triggers for Improved User Performance in 802.11 Wireless Networks," *Mobisys '06*, June 19-22, 2006.
9. IEEE 802.11 WG (2003) Draft Supplement to STANDARD FOR Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Specification for Radio Resource Measurement, IEEE 802.11k/D1.0. New York USA: The Institute of Electrical and Electronics Engineers, Inc.
10. V. Brik, A. Mishra and S. Banerjee, "Eliminating handoff latencies in 802.11 WLANs using Multiple Radios: Applications, Experience, and Evaluation", *ACM IMC*, Oct. 2005.
11. H. Velayos and G. Karlsson, "Techniques to Reduce IEEE 802.11b MAC Layer Handover Time," Kungl Tekniska Hogskolen, Stockholm, Sweden, Tech. Rep. TRITA-IMIT-LCN R 03:02, ISSN 1651-7717, ISRN KTH/IMIT/LCN/R-03/02.SE, April 2003.
12. "Darwin Streaming Server", <http://developer.apple.com/darwin/projects/streaming/>
13. "QuickTime Streaming", <http://developer.apple.com/documentation/QuickTime/RM/Streaming/StreamingClient/StreamingClient.pdf>.
14. N. Cranley, M. Davis, "Performance Evaluation of Resource Usage for Unicast Video Streaming over IEEE 802.11 WLAN Networks", In Proc. 5th Workshop on Applications and Services in Wireless Networks ASWN 2005, Paris, France, July 2005.
15. Jean Tourrilhes, "Wireless LAN resources for Linux," <http://www.hpl.hp.com/personal/JeanTourrilhes/Linux/Wireless.html>.
16. N. Cranley, M. Davis, "Performance Analysis of Network level QoS with Encoding Configurations for Unicast Video Streaming over IEEE 802.11 WLAN Networks", *WirelessCom 2005*, Maui, Hawaii, June 2005.
17. www.wireshark.org
18. www.linux.org
19. www.atheros.org
20. www.madwifi.org
21. www.skype.org
22. <http://netdude.sourceforge.net/doco/libpcapnav/>