Horizontal Handoffs within WLANs

A detailed analysis and measurement concerning voice like traffic

AJEET NANKANI

KTH Information and
Communication Technology

Master of Science Thesis
Stockholm, Sweden 2005

IMIT/LCN 2005-19

Horizontal Handoffs within WLANs: A detailed analysis and measurement concerning voice like traffic.

AJEET NANKANI < ajeet+msthesis@kth.se >

Last Updated: 31st July, 2005

Supervisor and Examiner:	Professor Gerald Q. Maguire Jr. < maguire@it.kth.se >
Industry supervisor:	Ulf Börjeson < ulf@r2m.se >
Sponsor:	R2M AB. < http://www.r2m.se >

School of Information and Communication Technology (ICT),
Royal Institute of Technology (KTH), Stockholm, Sweden.

Abstract

IEEE 802.11 based Wireless Local Area Networks (WLANs) in addition to being used as access networks for providing traditional data services, are now also being used as access networks for providing realtime services such as VoIP and multimedia streaming. These realtime services are sensitive to latency, hence requiring seamless or low delay service from the lower layers throughout an ongoing session. The IEEE 802.11 standard does not define any technique or algorithm to provide seamless connectivity during the process of handoff, hence it does not require 802.11 based WLANs to provide the same. Thus, it is typical that there is a latency of 500 milliseconds to 1000 milliseconds during the handoff, before the mobile station can connect and receive data from the new access point (AP). However, many realtime services can not tolerate this much latency.

The problem of handoff latency is further aggravated when WLANs are secured using IEEE 802.11i standard and when Authentication, Authorization & Accounting (AAA) services are involved in controlling network access to 802.11 based WLANs. This thesis will address the entire handoff process and examine the latency -- especially regarding AAA services. Different techniques and suggestions will be presented and analyzed closely at different layers and based on the results, an appropriate/efficient algorithm is suggested which will reduce this handoff latency, such that that seamless handoff can be achieved and realtime services can be provided over 802.11i enabled IEEE 802.11 WLANs.

Keywords: WLAN, 802.11, handover, handoff, 802.11i, VoIP over WLAN, horizontal handoff, horizontal handover, VoWLAN.

Sammanfattning

Wireless Local Area Network (WLAN), baserat på IEEE 802.11 har traditionellt nyttjats som som accessnät för vanliga datatjänster. Ett allt vanligare användningsområde har blivit att nyttja samma nät för realtidstjänster som Voice over IP (VoIP) och mutimedia. Realtidstjänster är känsliga för fördröjningar. Fördröjningar som bland annat kan erhållas från de lägre nivåerna i OSI-stacken. IEEE 802.11-standarden definierar ingen teknik eller algoritm för att säkerställa avbrottsfri/fördröjningsfri transmission av data vid handoff och följdaktligen så kan man idag inte luta sig mot denna standard för att erhålla denna funktionalitet. Med nyttjande av befintlig IEEE 802.11 standarder erhålls fördröjningar på mellan 0,5 till 1 sekunder. Detta är naturligtvis inte acceptablet för många realtid och realtidsliknande tjänster.

Problemet vid handoff accentueras ytterliggare om kravs ställs på AAA-tjänster för att säkerställa säkerheten i ett IEEE 802.11-baserat WLAN. Denna uppsats adresserar hela handoffprocessen med tillhörande fördröjningar – speciellt med hänsyn till AAA-tjänsterna. Olika tekninker och förslag presenteras och analyseras på olika nivåer. Baserat på erhållna resultat föreslås en algoritm för att reducera tidsåtgång vid handoff, så att realtidsliknande tjänster erhålls, utan störande fördröjningar, vid nyttjande av 802.11i.

Nyckelord: WLAN, 802.11, handover, handoff, 802.11i, VoIP over WLAN, horizontal handoff, horizontal handover, VoWLAN.

Acknowledgments

I would like to thank: Prof. Gerald Q. Maguire – for giving me opportunity to work under his supervision and giving me valuable feedback from time to time during the thesis work, Ulf Börjeson other staff at R^{2}M – for sponsoring this thesis work, providing working space in their office premises and motivating and encouraging me from time to time. I would like to mention also name of Jouni Malinen who is author of HostAP driver and others on the HostAP mailing list who answered many of my queries, without which it would have been difficult to do the test experiments.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Overview of the problem

During the past few years, there has been a tremendous growth in deployment of IEEE 802.11[1] based WLANs all over the world, basically due to ease of deployment, low cost, and operation in unregulated wireless frequency band. The biggest advantage WLANs provide to users is the flexibility to move around with their device while still being connected to an access network without the hassle of any wires, in the area covered by that WLAN. We call this pedestrian mobility.

Initially these WLANs were typically used only for data access, such as providing internet access to end stations, and users did not move so often while their devices were connected to WLANs. But recently the 802.11 based interface has become quite cheap and most PDAs, smartphones and other small portables now have an 802.11 interface built into, thus the end users of these nomadic devices which we will call here mobile stations (STA), have started to use realtime services like Voice over Internet Protocol (VoIP) and multimedia streaming on these nomadic devices through their WLAN interface. Due to nature of these realtime services, low latency is the basic requirement for these realtime services to be successfully used without any quality degradation.

The fact that 802.11 enabled devices have become very light in weight, enables users of these devices to move easily while still being connected via the WLAN within its covered area. However this movement of the users while still being connected to WLAN can result in a lot of handoffs, which we define as de-attaching from the current Access Point (AP) and attaching to another AP, both of which are connected with a distribution system(DS) (i.e., a backbone LAN).

When the STA is attached to an WLAN AP, there is no problem in using these realtime services in normal situations, but the problem arises only when the real time services are being used and STA makes an handoff. The handoff occurs when STA moves away from the current AP, which we call old Access Point (old-AP) to which it is attached currently, the STA tries to find another AP, which we call new Access Point (new-AP), and de-attaches from old-AP and attaches to the new-AP, so that from now on it receives data from new-AP. During the process of handoff there is a latency of a few hundred milliseconds to a second before the STA receives data from the new-AP. This handoff latency problem becomes more complex and the latency longer when WLANs are secured using IEEE 802.11i (see section 1.5) and AAA services are required for WLAN network access, which means the STA has to be authenticated by every AP it wishes to attach to, in order to gain access to network services. This means that during the handoff STA has to be re-authenticated to the WLANs generally through the new-AP before gaining access to any services through this new-AP, this re-authentication process increases the overall handoff latency to around seconds. The IEEE 802.11 published standard itself does not mention any technique or algorithm to reduce this handoff latency.

The thesis will examine the whole handoff process in very deep detail and based on that analysis I propose new algorithms and methods for reducing the total handoff delay, when WLANs are fully secured by technologies and methods which conforms fully with the IEEE 802.11i standard.

1.2 Overview of the Wireless Local Area Networks (802.11 WLANs)

In 1997 IEEE introduced, the first Wireless Local Area Network (WLAN) standard, IEEE 802.11[1]. The 802.11 family of standards define the communication protocols between wireless stations and the network access points that bridge wireless and wired networks. It was a beginning, but the standard had serious deficiencies. Initially 802.11 only supported speeds of up to 2 Mbps. It supported two entirely different methods of encoding, Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS), leading to confusion and incompatibility between equipment. It also had problems dealing with collisions and with signals reflected back from surfaces, such as walls. These defects were soon addressed, and in 1999, the IEEE 802.11b[2] standard arrived.

The 802.11 standard is similar in most respects to the IEEE 802.3 Ethernet standard, which can be seen in figure 1.1 .

IEEE 802.2 Logical Link Control (LLC)		OSI Layer 2 (DataLink)
IEEE 802.11 (MAC)	IEEE 802.3 (MAC)	
802.11 PHY	802.3 PHY	OSI Layer 1 (Physical)

Figure 1.1 - IEEE 802.11 and 802.3 standards mapped to the OSI reference model.

1.2.1 WLAN architecture

The 802.11 architecture is comprised of several components and services that interact to provide the station with connectivity and mobility. The main components are:

WLAN Station (STA) – The WLAN Station (STA) is the basic component of the wireless network. Such a STA is any device which provides 802.11 functionality and implements 802.11 medium access control (MAC) and physical layer (PHY).

WLAN Access Point (AP) – The WLAN Access Point (AP) is essentially a STA, but with additional functions to support bridging (i.e., layer 2 forwarding) and 802.11 management. The AP is usually connected to a wired network, and can relay data between devices on each side of the bridge.

Basic Service Set (BSS) – Basic Service Set (BSS) is the basic building block of an 802.11 WLAN. The BSS consists of a group stations which communicates directly.

Infrastructure Basic Service Set (IBSS) – An Infrastructure Basic Service Set is a BSS with a AP. The AP provides a local relay function for the BSS. All stations in the BSS communicate with the access point and no longer communicate directly. All frames are relayed between stations by the access point. This AP may be connected to a distribution system.

Distribution System (DS) – In typical cases large areas or the whole building premises needs to be covered with WLAN to provided access at every point in the required area. To achieve this more than one AP needs to be utilized to cover the whole area. To enable roaming between multiple access points and connections to wired network resources, the 802.11 standard specifies a DS, which provides wired interconnections between APs. This DS can be wired or wireless, and needs to fulfill certain functions. The 802.11 standard says that the distribution system may be of any technology, such as Ethernet, token ring, or any other network type however, the majority of actual installations, utilize Ethernet. The DS shown in figure 1.2 connects the APs, forwards traffic, and facilitates the movement of mobile STAs within a larger area.

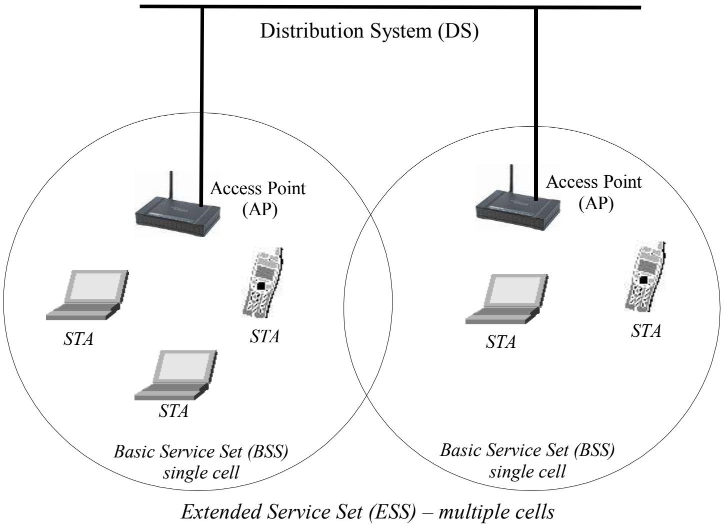


Figure 1.2 - Typical WLAN network architecture

Extended Service Set (ESS) – To allow a greater mobility of mobile STAs and not confine them to a single BSS, multiple infrastructure BSSs can be combined to form an Extended Service Set (ESS). The AP determines whether frames need to be relayed within the BSS, forwarded within the ESS via the DS or routed to outside entities. The DS hides the mobility of the stations by treating the whole ESS as a single network (L2 domain), which makes it possible to use existing link layer protocols in a locally mobile setting.

Service Set Identifier (SSID) – Service Set IDentifier (SSID) is a unique label that distinguishes one WLAN from another. All APs and STAs attempting to become a part of a specific WLAN must use the same SSID. Wireless STAs use this SSID to establish and maintain connectivity with APs.

Mobility – Mobility of wireless stations may be the most important feature of a WLAN. A WLAN would not serve much purpose other than reducing wiring costs if stations were not able to move about freely from location to location either within a specific WLAN or between different WLAN segments. As described above , the 802.11 MAC appears to the upper layers of the network just as if it were a standard Ethernet based LAN. Thus the 802.11 MAC layer is forced to handle station mobility in a fashion that is transparent to the upper layers of the IP stack. This forces many functionalities into the 802.11 MAC layer which are usually handled by upper layers.

1.3 IEEE 802.11 WLAN standards

This section is divided into two parts, the IEEE 802.11 standards which have been released/approved until now (July 2005), and the IEEE 802.11 standards which have not yet been approved and are being worked upon but could be released in the near future. These not yet approved 802.11 standards could enhance data rates, provide more functions, facilitate management, etc.

1.3.1 Approved IEEE 802.11 standards (July 2005)

802.11b

The IEEE 802.11b – 1999[2], extension of the original 802.11 standard[1] boosted wireless throughput from 2 Mbps all the way up to 11 Mbps. An 802.11b device can transmit up to 100m under good conditions, although this distance may be reduced by the presence of obstacles such as walls.

The 802.11b upgrade dropped FHSS in favor of DSSS. DSSS has proved to be more reliable than FHSS, and settling on one method of encoding eliminates the problem of having a single standard that includes two types of equipment that aren’t compatible with each other. However all 802.11b devices are compatible with older 802.11 DSSS devices, but are not compatible with 802.11 FHSS devices. Because 802.11b differs from standard 802.3 wired Ethernet only at OSI Layers 1 and 2 [page 3 of 1], hence it is inter-operable with standard wired Ethernet and has the same 802.2 Logical Link Control. Hence it looks like an Ethernet to nearly all applications.

Today 802.11b devices are the most widely available WLAN equipment, and currently are deployed in the largest number of installations around the world.

802.11a

The first amendment, IEEE 802.11a – 1999[3] uses the 5 GHz band which is a different band than the 2.4GHz band used by 802.11b. Because of its higher frequency, a different modulation technique and a larger bandwidth allotment was feasible. The 802.11a standard not only achieves speeds of up to 54 Mbps, but allows for a larger number of channels that do not overlap in frequency. IEEE 802.11a employs Orthogonal Frequency Division Multiplexing (OFDM) and provides 23 channels in total.

802.11g

Introduced in 2003, IEEE 802.11g[4], is an extension of 802.11b and operates in the same 2.4-GHz band as 802.11b. It increases data rates up to 54 Mbps using OFDM technology. Because 802.11g specifies backward compatibility with 802.11b, an 802.11b device can interface directly with an 802.11g access point and vice versa. Some newer 802.11b access points can even be upgraded to be 802.11g via relatively easy firmware upgrades.

802.11d

IEEE 802.11d[5] is a supplement to the Media Access Control (MAC) layer in 802.11 to promote worldwide use of 802.11 WLANs. It allows access points to communicate information about the permissible radio channels with acceptable power levels for user devices. Because 802.11 standard devices cannot legally operate in some countries, the purpose of 802.11d was to add features and restrictions to allow WLANs to operate within the rules of these countries. These variations include allowed frequencies, allowed power levels, and allowed signal bandwidth. The specification eliminates the need for designing and manufacturing dozens of different hardware version, each for use in a particular jurisdiction. The 802.11d specification is thus well suited for systems that want to provide global roaming, which essentially means that the same mobile station can be used in different regulatory domains.

802.11h

IEEE 802.11h[6] was introduced in 2003 to address the requirements of the European regulatory bodies. Since the IEEE 802.11a WLAN standard operates in 5GHz band and in some European countries some military RADAR systems and medical devices also operates in same band, to resolve interference introduced by IEEE 802.11a WLAN, 802.11h was introduced to address these issues. It provides dynamic channel selection (DCS) and transmit power control (TPC) for devices operating in the 5GHz band (802.11a).

802.11j

IEEE 802.11j[7]was introduced in October 2004 to address requirements of the European regulatory body. The main intent of 802.11j is to add channels in the radio frequency band of 4.9 GHz to 5.0 GHz. In addition, certain changes are made, to satisfy Japanese legal requirements concerning wireless transmitter output power, operational modes, channel arrangements, and spurious emission levels.

802.11c

IEEE 802.11c provides the information required to ensure proper bridging operations, when APs are deployed.

802.11F

IEEE 802.11F which is also called the Inter Access Point Protocol (IAPP) is a recommended practice (RP). It describes an optional extension to the IEEE

802.11 standard that enables communications between APs of *different* vendors. Basically it defines a protocol for communication between APs in order to facilitate roaming. It does so by transferring context information related to a STA from one AP to the other AP where STA has roamed to. Even though 802.11F was defined to facilitate the roaming process in scenarios where VoIP applications are used it in fact delays the roaming process in most cases.

802.11i

IEEE 802.11i [8](also known as WPA2) is an amendment to the 802.11 standard specifying security mechanisms for 802.11 WLANs. The standard was ratified on 24 June 2004, and supersedes the previous security specification, Wired Equivalent Privacy (WEP), which was shown to have severe security weaknesses. Wi-Fi Protected Access (WPA) is a subset of 802.11i. It had previously been introduced by the Wi-Fi Alliance as an intermediate solution to WEP insecurities. Unlike WEP and WPA which uses only an RC4 stream cipher, 802.11i make use of the Advanced Encryption Standard (AES) block cipher. Functional aspects and the different components on which 802.11i builds are discussed in detail in section 1.4 and 1.5.

1.3.2 Ongoing IEEE 802.11 standards activity (July 2005)

802.11e

IEEE 802.11e provides Quality of Service (QoS) support for WLAN applications, which will be critical for delay-sensitive applications such as Voice over IP over WLAN (VoWLAN). The standard will provides classes of service with managed levels of QoS for data, voice, and video applications. 802.11e enhances the 802.11 Media Access Control layer (MAC layer) with a coordinated time division multiple access (TDMA) construct, and adds error-correcting mechanisms for delay-sensitive applications such as voice and video. The 802.11e standard is expected to be ratified this summer (2005) and should start appearing in products this fall (2005). Already, many vendors have deployed parts of it based on the draft standard.

802.11k

IEEE 802.11k is a proposed standard for how a WLAN should perform channel selection, roaming decisions, and transmit power control (TPC) in order to optimize network performance. 802.11k aims to provide key client feedback to APs and switches. It is focused on standardizing the radio measurements that will allow uniform measurement of radio information across different manufacturer platforms. The proposed standard defines a series of measurement requests and reports, that detail Layer 1 and Layer 2 client statistics. In most cases, APs or WLAN switches ask STA clients to report data, but in some cases client STAs might request data from APs. Some of the measurements 802.11k defines are:

- Roaming decisions,
- RF channel knowledge,

- Hidden nodes,
- Client statistics, and
- Transmit Power Control (TCP).

To improve roaming decisions, APs or WLAN switches can provide a site report to clients. The standard defines a beacon request, in which an AP asks a client to go to a specific channel and report all the AP beacons it hears. The AP collects the data, and it or a WLAN switch will analyze the beacon information, looking at details such as what services and encryption types each AP supports and how strongly the client heard the AP. Then the switch or AP generates an ordered list of APs, from best to worst service, called the site report. This report when request by client STA from APs would certainly help. the STA to make a better roaming decision and could be able to do a fast handoff. Figure 1.3 shows how site report would help in roaming decision.

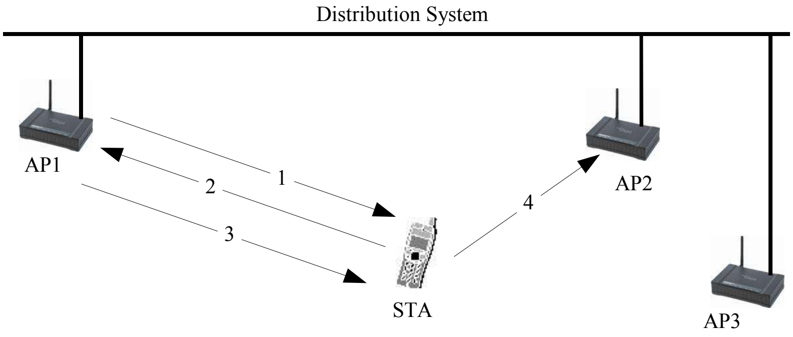


Figure 1.3 - Possible scenario showing how the site report influences the roaming decision.

Possible message flows between APs and the Cleint STA during handoff as shown in figure 1.3

1. *When th AP1 determines that the client STA connected to it is moving away from it, AP1 informs the client STA to prepare to move to another AP.*
2. *The client asks AP1 to provide it with the list of preferred APs nearby.*
3. *The AP1 responds with a site report, which contains list of its preferred neighbor APs.*
4. *The client STA immediately switches to the channel of the best AP listed in the site report and connects to it. This contributes to faster and perhaps even uninterrupted wireless connection.*

Currently, APs and clients cannot share channel information. With 802.11k, an AP could have a client build a "noise histogram," which will display all non-802.11 energy in that channel. An AP also can request data about channel load or how long the channel was used during a given time. An AP or WLAN switch will then know if there's too much interference or traffic on a channel to use it.

Hidden nodes are clients or APs that other clients or APs cannot hear. In 802.11, nodes listen to the airwaves before transmitting to avoid collisions. When a hidden node is present, multiple nodes can transmit simultaneously, creating interference that degrades WLAN performance. With 802.11k, clients track hidden nodes and APs query clients for these lists. This information tells

APs about clients on the edge of their cells. APs can use this information to direct clients to APs from which they might get better service.

Client statistics are limited today to statistics that APs or WLAN switches maintain. Today's WLANs track items such as retries, packets transmitted and packets received. With 802.11k, APs and WLAN switches can query all clients to get reports on their statistics. With both data sets, a WLAN system will have a more complete view of network performance hence would be in a position to make decisions to enhance the overall throughput. For example in a network conforming to 802.11k, if the AP having the strongest signal is loaded to its full capacity, a new wireless device would be told to connect to one of the underutilized APs. Even though the signal may be weaker, the overall throughput is greater because more efficient use is made of the network resources.

TPC was defined in 802.11h to meet regulatory requirements in the 5-GHz band in Europe. With 802.11k, extends the use of TPC procedures in other regulatory domains and frequency bands to reduce interference and power consumption, and provide range control.

802.11m

802.11m performs editorial maintenance, corrections, improvements, clarifications, amendments, and interpretations relevant to documentation for 802.11 family specifications. The term 802.11m also refers to the set of maintenance releases itself.

802.11n

Task Group N (802.11 TGn) was formed in July 2003, to define high throughput enhancements. TGn's objective is to define modifications to the Physical Layer and Medium Access Control Layer (PHY/MAC) that deliver a minimum of 100 Mbps throughput at the Media Access Control Layer-Service Access Point, while the Over The Air (OTA) throughput would be 200+ Mbps. This minimum throughput requirement represents an approximate 4x leap in WLAN throughput performance compared to today's 802.11a/g networks. At the same time, 802.11n expects a smooth adoption transition by requiring backward compatibility with existing IEEE WLAN legacy solutions (802.11a/b/g). 802.11n would achieve such high throughput by using an advanced technology called Multiple Input Multiple Output (MIMO) or smart antenna systems. MIMO exploits the use of multiple signals transmitted into the wireless medium and multiple signals received from the wireless medium to improve wireless performance.

802.11p

IEEE 802.11p addresses the wireless access in vehicular environments (WAVE). IEEE 802.11p expands on conventional 802.11 wireless networking to allow for provisions that are specifically useful to automobiles. It would provide a more advanced handoff scheme, mobile operation, enhanced security, identification, peer-to-peer (ad hoc) authentication, and most importantly, communications in the automotive allocated 5.9 GHz spectrum. 802.11p will be used as the basics for automotive-targeted communications or Dedicated Short Range Communications (DSRC), which is a general purpose communications link between the vehicle and the roadside (or between vehicles) using the 802.11p protocol. The 802.11p protocol improves on the range and speed of transmission on the dedicated 5.9 GHz licensed band, promising around 300m and 6 Mbit/s on average. The typical usage scenarios are toll collection, vehicle safety services, and commerce transactions via cars. The 802.11r Task Group is working on reducing the handoff latency when client devices transition between APs or cells in an Extended Service Set comprising APs in the same network.

802.11r

IEEE 802.11r will provide standards for fast roaming, quick re-association with a new AP in an ESS after a client STA moves out of the range of an AP in the same ESS to which it is currently associated. The standard will seeks to foster the use of mobile, wireless VoIP phones and other time-sensitive WLAN applications by eliminating perceptible disconnections which usually occur during handoff. This will make it easier to use VoIP (VoWLAN) and other real-time interactive applications and will facilitate the deployment of VoWLAN portable phones. The new 802.11r roaming methodology must also support the 802.11i security protocols. These protocols may involve lengthy authentications with the RADIUS server. These authentications, in turn, have the potential to severely impact the roaming speed. The 802.11r task group is studying new protocols, which may use *pre-authentication* as defined in 802.11i. Such protocols promise to achieve the desired roaming performance even when lengthy 802.11i authentication is required.

802.11s

This IEEE task group is defining standards for wireless mesh routing. Although mesh networks are already in use for very large deployments in cities, and in some industry sectors, none of these systems inter operate or are suitable for domestic or office environments.[9] Thus 802.11s will focus on an infrastructure mesh standard to allow 802.11 APs from multiple manufacturers to self-configure into multi-hop wireless topologies, thus forming a multi-hop wireless mesh network which would be adaptive (dynamic path configuration), self-configurable (automatic topology learning), and fault tolerant (having no single point of failure). Such networks will extend WLAN range by allowing data to pass through wireless nodes providing coverage beyond the typical WLAN connectivity limit of 100m. Example usage scenarios for mesh networks include interconnectivity for devices in the digital home, large campuses, community area networks or hot zones and hard-to-wire areas. The standard is expected to be designed to be extensible by manufacturers to enable diverse usage scenarios with differing functional requirements. For example, some applications may require quick ad hoc setup and tear down of a mesh while others require large scale and maximum throughput.

802.11.2 (802.11T)

The testing of 802.11 devices and systems for performance and stability is a challenge for the industry. The 802.11 protocol's complexity brings with it a corresponding test complexity that is further compounded by the prevalence of RF interference and the inherent mobility of the wireless devices. RF interference makes it difficult to obtain repeatable results that can be correlated among multiple laboratories. To address the need for standard testing requirements and performance evaluation 802.11.2 will be released as Recommended Practice (RP) thus providing a set of performance metrics, measurement methodologies, and test conditions to enable manufacturers, test labs, service providers, and users to measure the performance of 802.11 WLAN devices and networks at the component and application level. One thing to note is the changing in the name of this RP which is different from usual naming series in 802.11 family. Here 802.11T is the Task Group which will release IEEE 802.11.2 Recommended Practice.

802.11u

The Wireless Internetworking with External Networks (WIEN) Study Group , 802.11u, is establishing standards for the integration of 802.11 and other wireless systems such as 3G cellular. The task group is studying access router identification, MAC address anonymity, scalability, policy enforcement, access control, quality of service, and billing administration; in addition to the other requirements for interoperation between network systems. To achieve this 802.11u takes 802.11 related output from 802.21[10] and will make the required amendments in 802.11 standard, which will be known as 802.11u.

802.11v

The 802.11v study group focuses on standards for wireless network management with the goal of providing a complete and coherent upper layer interface for managing devices in 802.11 wireless networks. Problems with traditional SNMP in the distributed WLAN environment are being addressed. The 802.11v group is dependent on the 802.11k group, which is defining measurements that will be incorporated into the management interface being defined by 802.11v. IEEE 802.11v will allow APs, switches, and client STAs to communicate and cooperate among themselves so that WLANs can do self-tuning and dynamically adjust conditions. Such coordination will play a big role in reducing interference, increasing throughput, and helping to improve overall WLAN reliability.

802.11w

IEEE 802.11w will define enhancements to 802.11 Medium Access Control (MAC) layer to provide, as appropriate, mechanisms that enable data integrity, data origin authenticity, replay protection, and data confidentiality for selected IEEE 802.11 management frames including, but not limited to: action management frames, de-authentication and disassociation frames.

Table 1.1 gives overall summary of the IEEE 802.11standards, amendments and recommended practice (RP) both released and unreleased. The table also specifies the predicted release time[11] for unreleased IEEE 802.11standards.

Standard	Description	Status
802.11	Base standard 2.4 GHz - IR, DSSS and FHSS	Completed
802.11a	5 GHz OFDM 5Mb/s	Completed
802.11b	2.4 GHz DSSS 11Mb/s	Completed
802.11c	Bridge Operations	Completed
802.11d	Global Harmonisation	Completed
802.11e	MAC Enhancements for QoS	Sep 2005
802.11F	Inter Access Point Protocol	Completed
802.11g	2.4 GHz DSSS and OFDM 54Mb/s	Completed
802.11h	Spectrum and Transmit Power Management	Completed
802.11i	MAC Enhancements for Enhanced Security	Completed
802.11j	Japanese Regulatory Extensions to 802.11a	Completed
802.11k	Radio Resource Measurements	Jun 2006
802.11m	Standard Maintenance	Mar 2006
802.11n	Enhancements for High Throughput	Dec 2006
802.11p	Wireless Access for the Vehicular Environment	Sep 2006
802.11r	Inter-AP handoffs (Fast Roaming)	Sep 2006
802.11s	ESS Mesh Networking	Jun 2008
802.11.2*	Wireless performance prediction for testing	Dec 2007
802.11u	InterWorking with External Networks	Sep 2007
802.11v	Wireless Network Management	Sep 2007
802.11w	Protected Management Frames	Dec 2007

Table 1.1 - Summary of 802.11 standards

** 802.11.2 will be released as a recommended practice (RP) by 802.11T Task Group*

1.4 Secure WLAN Infrastructure

A primary concern when installing commercial wireless networks is security. The rapid growth and popularity of wireless networks in both the commercial and residential market led to the use of wireless for many diverse applications, including the transmission of private information.

All flavors of the 802.11 WLAN standard includes a security protocol called Wired Equivalent Privacy (WEP), which encrypts data packets well enough to keep out causal eavesdroppers. WEP encrypts each 802.11 packet separately with an RSA RC4 cipher stream generated by a 64-bit RCA key.

But several cryptoanalysts have identified weaknesses in the RC4's key scheduling algorithm that make the network vulnerable to hackers. Software tools such as AirSnort have already been developed to enable hackers to crack WEP and gain access to the WLAN.

To rectify WEP vulnerability, IEEE started developing a more secure alternative named IEEE 802.11i [8] standard. However WLANs were being deployed everywhere, thus need to have strong secure alternative to WEP before IEEE 802.11i was released was very high, so the Wi-Fi Alliance in conjunction with IEEE introduced an enhanced security scheme called Wi-Fi Protected Access (WPA) as an alternative to WEP in the first quarter of 2003.

Wi-Fi Protected Access is a specification of a standards-based, interoperable security enhancement that greatly increased the level of a data protection and access control for existing and future wireless LAN systems. Designed to run on existing hardware as a software upgrade, Wi-Fi Protected Access is derived from and is forward compatible with the IEEE 802.11i standard which was released in June 2004.

The main components of 802.11i are the data-confidentiality protocol Counter-Mode/CBC-MAC Protocol (CCMP) and IEEE 802.1X's key-distribution system to control access to the network. Because IEEE 802.11 handles unicast and broadcast traffic differently, each traffic type has different security concerns. With several data-confidentiality protocols and the key distribution, IEEE 802.11i includes a negotiation process for selecting the correct confidentiality protocol and key system for each traffic type. Other features introduced include key caching and pre-authentication which are described later in this section.

There are many alternatives to 802.11i, some of which are explained by J-O Vatn in his dissertation [12], but here we only focus on the IEEE 802.11i standard.

1.5 Functional aspects of IEEE 802.11i

1.5.1 Security parameters negotiation between STA and AP

In 802.11i enabled WLANs security parameters between AP and STA are negotiated using beacon, probe response, and (re)association frames. The IEEE 802.11i enabled APs sends Robust Security Network Information Element (RSNIE) in their beacons and probe response frames. This RSNIE containes information about security features and cipher suites supported by AP. Based on its own security policy the STA selects security features and cipher suites from APs RSNIE and constructs its own RSNIE which STA sends in (re) association frames. This negotiation of security parameters is later validated during 4-way handshake described later.

1.5.2 Role of IEEE 802.1X in IEEE 802.11i

IEEE 802.1X provides a framework to authenticate and authorize devices connecting to a network. It prohibits access to the network using controlled port technique as shown in figure 1.5 until such devices pass authentication. IEEE 802.1X also provides a framework to transmit key information using EAPoL-key exchange between authenticator and supplicant.

IEEE 802.1X has three main entities as shown in Figure 1.4

- Supplicant
- Authenticator
- Authentication server

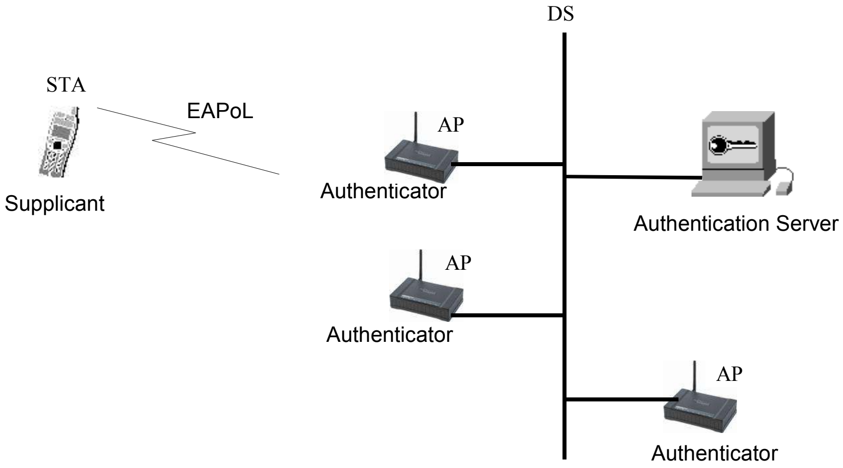


Figure 1.4 - System Architecture of 802.11i enabled WLAN

As shown in figure 1.4, the AP takes the role of the authenticator and the STA (client) the role of supplicant. The supplicant authenticates with the authentication server through the authenticator. In IEEE 802.1X, the authenticator enforces authentication. However, the authenticator doesn't need to do the authentication, instead the authenticator forwards authentication traffic between the supplicant and the authentication server.

Between the supplicant and the authenticator, the protocol is IEEE 802.1X which is basically a EAP challenge response paradigm. EAP uses four types of messages: EAP-Request, EAP-Response, EAP-Success and EAP-Failure. The authenticator(AP) sends the challenge (EAP-Request) when the supplicant is (re)associated with the AP and waits for a response from supplicant (EAP-Response). After several messages to exchange the credentials, the supplicant will receive an EAP-Success/Failure message. The authentication mechanisms between the authentication server and supplicant can be any EAP based methods; which includes EAP-TLS, EAP-SIM, EAP-MD5, EAP-CHAP, EAP-TTLS, etc. Of all these EAP methods, EAP-TLS is most widely used and is considered as the de facto standard for WLAN security in the 802.11i standard.

The protocol between the authenticator and authentication server isn't defined in IEEE 802.1X or IEEE 802.11i. However, RADIUS or DIAMETER is typically used between authenticator and authentication server.

As many messages are exchanged during the authentication process at each AP it adds considerable delay until a STA can start to receive data traffic from the new AP.

The notion of controlled and uncontrolled ports is used to distinguish between 802.1X and non 802.1X traffic as shown in figure 1.5. The uncontrolled port only passes authentication traffic i.e., (802.1X EAP packets) between the supplicant and the authentication server.

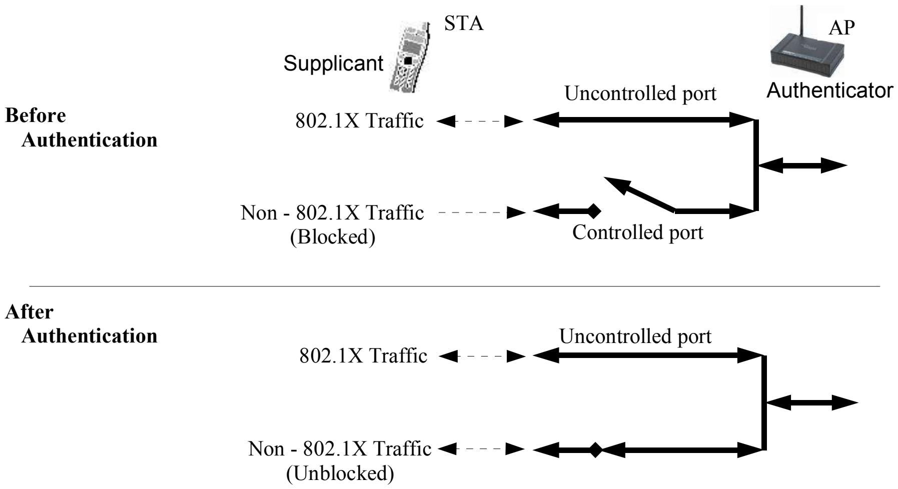


Figure 1.5 - Notion of controlled and uncontrolled ports in 802.1X

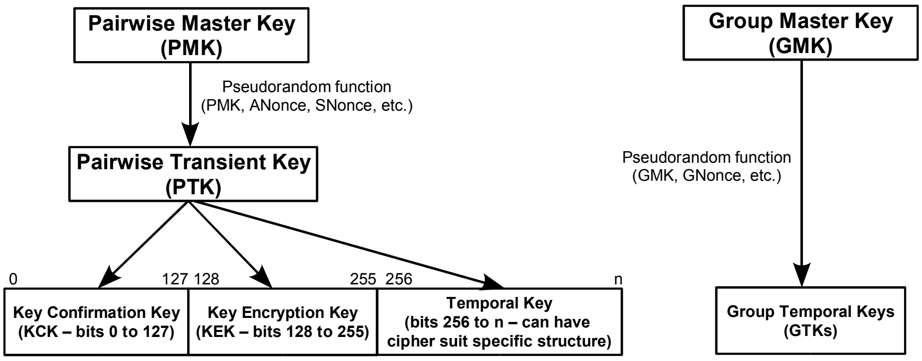
Once the authentication server concludes authentication with the supplicant, the authentication server informs the authenticator of the successful authentication and passes the keying material to the authenticator. At this point, the supplicant and the authenticator share key material through an EAPoL-key exchange. If all exchanges have been successful, the authenticator can now allow traffic to flow through the controlled port, giving the STA to access to the network.

1.5.3 Key hierarchy

The IEEE 802.11i EAPoL-key exchange uses a number of keys and has a key hierarchy to divide up the initial key material into useful keys. The two key hierarchies are:

- Pairwise key hierarchy and
- Group key hierarchy

Both hierarchies are shown in figures 1.6. These keys get used in the EAPoL-key exchanges. IEEE 802.1X defines an RC4 EAPoL-key frame. However, IEEE 802.11i defines its own EAPoL-key exchanges. In the IEEE 802.11i specification, these exchanges are referred to as the 4-way handshake and the group key handshake.



Pairwise Master Key (PMK)
Pseudorandom function (PMK, ANonce, SNonce, etc.)
Pairwise Transient Key (PTK)
0
127 128
255 256
n
Key Confirmation Key (KCK – bits 0 to 127)
Key Encryption Key (KEK – bits 128 to 255)
Temporal Key (bits 256 to n – can have cipher suit specific structure)
Group Master Key (GMK)
Pseudorandom function (GMK, GNonce, etc.)
Group Temporal Keys (GTKs)

Figure 1.6 - Pairwise Master Key (PMK) and Group Key hierarchy

1.5.4 EAPOL-key exchanges

Two main EAPOL-key exchanges are defined in IEEE 802.11i. The first is referred to as the 4-way handshake and the second is the group key handshake.

4-way handshake

The 4-way handshake does several things:

- Confirms the PMK between the supplicant and authenticator
- Establishes the temporal keys to be used by the data-confidentiality protocol
- Authenticates the security parameters that were negotiated (AP's RSNIE and STA's RSNIE)
- Performs the first group key handshake
- Provides keying material to implement the group key handshake

The reason it's called the 4-way handshake is because four packets are exchanged between the supplicant and the authenticator as shown in figure 1.7 and described below.

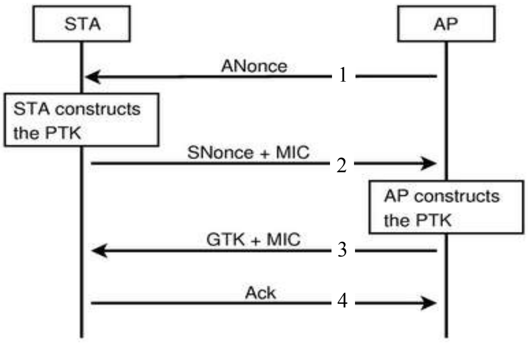


Figure 1.7 - 4 way handshake in 802.11i

- *4-way handshake message 1*

  In the first message, the AP sends the STA nonce, which is referred to as the ANonce.

- *4-way handshake message 2*

  The STA creates a nonce, which is referred to as the SNonce. The STA can now calculate the (Pairwise Transient Key) PTK. In the second message, the STA sends the SNonce along with Message Integrity Code (MIC – to prove , data origin authenticity) to the AP. The STA also sends the security parameters (RSN Information Element) that it used during association or re-association. The entire message gets an authentication check using the EAPoL - Key Confirmation Key (KCK) from the pairwise key hierarchy. The AP can then verify that the information, including the security parameters sent at (re)association, are valid.

- *4-way handshake message 3*

  In the third message, the AP sends the STA the security parameters (RSN IE) that it is sending in its beacons and probe responses. The AP also sends the Group Temporal Key (GTK) encrypted using the Key Encryption Key (KEK). Again, the entire message gets an authentication check (MIC), which allows the STA to verify that the information, such as the APs security parameters, are valid.

- *4-way handshake message 4*

  In fourth message the STA sends a confirmation to the AP, which indicates that the temporal keys are now in place and ready to be used by the data-confidentiality protocols.

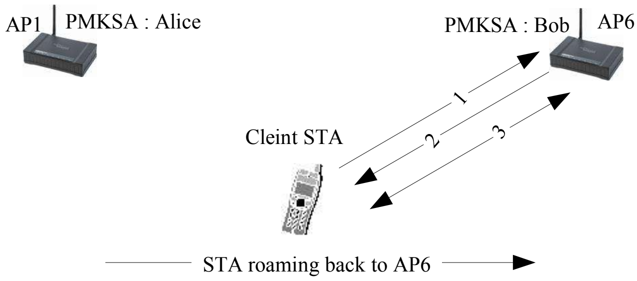
Group key handshake

The GTK used in the network may need to be updated due to the expiry of a preset timer. When a STA leaves the network, the GTK also needs to be updated. This is to prevent the STA from receiving any new multicast or broadcast messages from the AP.

To handle the updating, 802.11i defines a Group Key Handshake that consists of a two-way handshake, which consists of two messages. In first message the AP sends the new GTK to each STA in the network. The GTK is encrypted using the PTK assigned to that STA and is protected from being tampering using a MIC. In second message the STA acknowledges the new GTK and replies to the AP.

1.5.5 PMK Caching

As it has been discussed earlier that the main benefits WLAN provide over a traditional LAN is mobility in small as well as large areas (the later covered by many APs connected through DS), but during mobility in large areas the STA will change the AP to which it is attached and this requires it a do full 802.11i authentication (if the WLAN is secured using 802.11i). In scenarios where STAs move back and forth often between APs, this full 802.11i authentication can have a negative effect on the system performance. Because of this type of back and forth mobility a STA has to re-authenticate; which means STA has to start the whole 802.11i authentication process again and again with the same AP to which it has already been authenticated before. To address this issue and provide fast roaming support, 802.11i defines a method where an AP and STA can store previous security associations in their memory. The basic concept behind this key caching is for a STA and AP to retain a security association (SA) when a STA roams away from an AP. When the STA roams back to this AP, the security association can then be restarted as shown in figure 1.8 below.



1. *Association Request (PMKSA : Alice, PMKSA : Bob)*
2. *Association Response (success)*
3. *4-way handshake process*

Figure 1.8 - Usage of PMKSA caching

This key caching feature in 802.11i enabled WLANs on one hand reduces the load on the authentication server by not contacting the authentication server again for re-authentication and on the other hand it also reduces the time required to connect to the network, which actually is multi-fold improvement in total handoff latency.

To achieve key caching, IEEE 802.11i names the PMK Security Association (PMKSA) as shown in figure 1.8. This PMKSA caches PMK and not other keys in the key hierarchy, hence STA still needs to do 4-way handshake after re-association. As shown in figure 1.8 when a STA returns to an AP6 to which it has previously attached to, then STA sends the key name in the re-association request (message 1 in figure 1.8) that it sends to the AP6. The STA can send more than one key name in the association request. Note that STA also needs to know RSNIE of AP6 to send re-association request to AP6, as explained in section 1.5.1. If the AP sends success in the association response (message 2 in figure 1.8), then the STA and AP proceed directly to the 4-way handshake. The first message of the 4-way handshake will contain the name of the PMK security association. The 4-way handshake confirms that the STA and AP have the same PMK security association.

1.5.6 Pre-authentication in 802.11i

PMKSA caching reduces the the total 802.11i authentication time to just the time required for 4-way handshake, but still it requires that the STA has successfully associated with this AP before. To provide better support for fast roaming a way was needed to establish a PMKSA with a AP in the same DS although a STA had not yet been associated to that AP. This pre-authentication feature in 802.11i enables a STA to establish a PMKSA with an AP to which it will only associate with in the future.

The first time a STA associates with a AP in the DS, it must do a full 802.11i authentication. However, if the STA knows where it will be roaming, the STA can pre-authenticate to a new AP using the current AP and the backbone DS, as shown in figure 1.9.

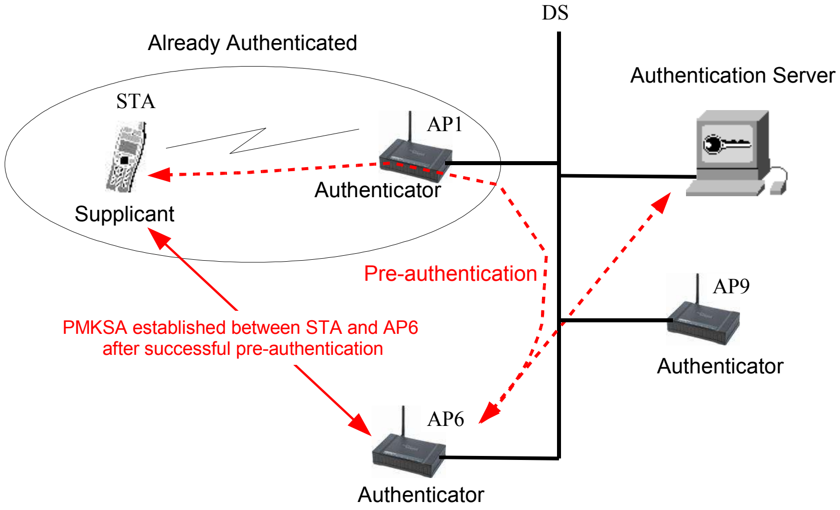


Figure 1.9 - Pre-authenticaion process in 802.11i enabled WLANs

The STA performs an authentication with the authentication server through the new AP, which acts as the authenticator. The pre-authentication packets traverse the existing AP via the DS to the new AP. Once the authentication is successful, the pre-authentication completes with a PMKSA established between the STA and the new AP. Pre-authentication then completes and doesn't perform the 4-way handshake with the new AP. When the STA roams to the new AP, it performs the same steps as in PMKSA caching. The STA sends the PMKSA name to the new AP in the association request. If the AP sends an association response with success, the STA and AP proceed directly to the 4-way handshake.

IEEE 802.11i states that STA can only start pre-authentication if the STA knows that new-AP supports pre-authentication, and this information is only available in RSNIE of AP [section 8.4.6.1 of 8]. Therefore to start this pre-authentication process STA needs MAC address (BSSID) of new-AP to communicate with it and its RSNIE to know about its pre-authentication capability. Also note that STA will also need this RSNIE just before roaming to the new-AP (see section 1.5.1) and later during 4-way handshake (see section 1.5.4). IEEE 802.11i does not describe how to get that BSSID and RSNIE of a new AP while connected to current AP, so currently we can only get it when the new AP is also in range of STA (so that beacon or probe response frames can be received from new AP). So there are still lot of opportunities to optimize the 802.11i pre-authentication process, such as ways get this RSNIE earlier.

Pre-authentication provides a way to establish a PMKSA before a STA associates. The advantage is that the STA reduces the time during which it is disconnected from the network. However, pre-authentication has some limitations and needs more processing power at APs and at authentication servers because STAs performing pre-authentication will add a load to the authentication server. Also since pre-authentication is done at the IEEE 802 layer, it doesn't work across IP subnets.

1.6 Voice over IP

Traditionally voice is carried over the Public Switching Telephone System (PSTN) which is a circuit switched system in which a circuit is maintained for the duration of a conversation, thus, a large part of the telephone network resource, wasted at any given time thus bandwidth in not used efficiently. In contrast, using data networks to deliver voice not only avoids the need for two separate systems for data and voice, but also makes better use of the network's resources.

As enterprises have already built data networks for their data communication needs in mid nineteen nineties there were initiatives from academics and industry to carrying voice over these packet switched data networks. However, these packet switched data networks were not designed for real time content delivery(e.g.,voice or video), hence initially they did not provide toll quality voice.

During the last decade, as bandwidth became cheaper and Voice over IP (VoIP) technology matured, VoIP became the attractive as it provided toll quality voice and integrating many advanced data services with voice. Thus enterprises started to use their data networks to carry their voice traffic in addition to their data traffic. During the last few years more and more home users around the world have also started to use the internet for their long distance voice calls. Today this VoIP technology has matured to such an extent that in different countries of the world there is tremendous growth in the number of operators offering broadband VoIP solutions to home and enterprise users.

IDC[1] forecast that the number of U.S. subscribers for residential VoIP services will grow from 3 million in 2005 to 27 million by the end of 2009[13]. Although it has been slow to develop in the United States and elsewhere, VoIP is finally beginning to show its potential in the consumer market.

1.7 Voice over IP over WLAN

IP applications do not care about underlying access network, method, or technology as long as the network provides IP connectivity with the required communication quality. Similarly VoIP does not care as long as underlying access network provides continuous IP connectivity sufficiently with low latency and acceptable packet loss rates. This means that Voice over IP over WLAN traditionally known as Wireless VoIP or VoWLAN, is the same in most cases as wired VoIP or simply VoIP because both wired access networks (mostly Ethernet and other broadband networks) and WLAN access networks provide IP connectivity with acceptable quality for VoIP applications. If the client STA attached to an AP in a WLAN is not mobile then VoIP applications

[1] http://www.idc.com

on that STA can be used as just like as if the STA was attached to a wired network as long as that wireless link provides basic quality of service needed by VoIP application.

So what is the big deal about VoWLAN?

The answer is that the quality of wireless link in a WLAN fluctuates significantly as it depends on many factors including, movement of client STA; interference from neighboring APs, STAs, and other electronic devices operating in same radio frequency range; and the number of users per AP - among other factors, so in many cases extra measures have to be taken to provide smooth VoWLAN. Furthermore the problem which is never encountered in a fixed wired network is the mobility of the client STA, that is when the client STA moves away from current AP in WLAN and attaches to another one then this transition from one AP to another one needs to be fast and efficient in order to maintain required link quality for VoIP applications over the transition time.

To address these short comings an IEEE 802.11 WG is working on many amendments to the IEEE 802.11 standard, which will enable smooth operation of VoIP applications on WLAN among other services.

How, today many large enterprises and campuses already have large deployments of WLANs and want their VoIP systems to be merged with WLANs to create a VoWLAN solution even before the amendments are released by the IEEE 802.11 WGs, They want to do so decrease their operation and maintenance cost.

Many commercial vendors want to capture this opportunity hence they have already introduced proprietary systems to support smooth operation of VoWLANs. This thesis will explore non-proprietary means of providing fast handoffs.

2 Background

2.1 Handoff process

In general handoff is defined as the process required to be executed to transfer the physical layer connectivity of a STA from one access point (AP) to another access point (AP). In addition to physical connectivity it may also a required transfer of some context or state information with respect to this STA. In terms of 802.11 WLAN this physical connectivity is called the association of an STA with the AP. Sometimes association is also called attachment.

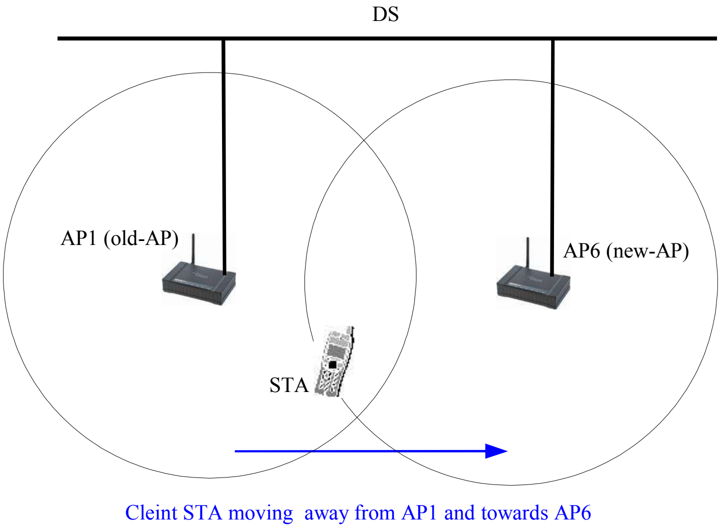


Figure 2.1 - Possible Handoff Scenario in WLAN networks

IEEE 802.11standard does not talk about handoff at all rather it assumes it to be same process when STA uses the first time it starts searching for APs, which in fact is what STA do after starting handoff process. But this notion misses the additional point that just before handoff there is a detection phase (as described in section 2.1.1) which significantly influences the performance of the overall handoff process.

Here we will describe the handoff process from the point when a STA operating in infrastructure mode has already associated with a WLAN AP in a Distribution System. When a STA has already associated to a current AP (which we will call a old-AP) and starts moving away from it as shown in figure 2.1, the wireless link quality between STA and this old-AP starts to deteriorate and at some point falls below threshold, which in turn triggers the handoff process to start. At this stage the STA starts to search for other APs to attach to by doing a scanning process(which can be either active or passive). Once the STA finishes scanning it sorts out the scan results and selects an appropriate AP to attach to and then STA authenticates (i.e., 802.11 authentication) and re-associates with the selected AP, by sending appropriate messages/frames to selected AP.

2.1.1 Handoff Trigger

Handoff related experiments can be done in deployed WLAN networks, in which one mobile client STA moves around the WLAN coverage area and follows a path where several monitoring stations are listening on each channel. These monitoring station will capture all traffic to and from the mobile client (STA). That traffic can be later analyzed to extract handoff related data and measure the delay. Mishra et al. [14] used this technique, but this procedure is quite cumbersome and requires lot of hardware and synchronization; hence it is unfeasible in most situations. Thus a WLAN testbed for the handoff related experiments is built in a small area roughly (equal the size of a single room) but in this case client (STA) gets strong signal from one AP in the whole room area where testbed is built, hence it won't do a handoff in that area even if the client (STA) is moved far away from the AP wihtin that room. So one must use some other technique to force the client the (STA) to do a handoff.

There are several methods to trigger the handoff, one which has been used most often and also used by Velayos et al. [15] is to turn off the radio of the AP to which the client (STA) is currently associated, the result will be that client will detect an abrupt loss of connectivity and hence try to search for new APs, and thus will connect to another AP present in the testbed. Even though this techniques simulates a handoff, it is not close to a real handoff in an actual WLAN, thus it can give incorrect and misleading measurements.

The other technique which is to reduce the output power of the AP to which the client (STA) is currently connected, the result would be that the client (STA) sees lower signal strength hence try to find an other AP and do a handoff once it finds an other AP in the testbed. Vatn [16] used this technique in his work, but an abrupt decrease in output power of the AP results in an abrupt decrease in signal strength which is not the case in real environment where signal strength degrades gradually when the client (STA) moves away from the AP to which it is connected currently. So this technique provide results which sometimes may not be equivalent to actual handoff results.

A better technique is to reduce the output power of all APs and the client (STA) to such an extent that the client (STA) will do a handoff to an other AP, even in a small testbed, when the client moves away from the AP to which it is already connected. Thus by physically moving a client back and forth between two APs in the testbed we can do a handoff which is closer to the actual handoff then any other technique used in testbeds in previous work in this area. To achieve this either built-in mechanisms in the APs and client STA could be used to reduce output power, but ideally one can use an attenuator between the AP and antenna, for example a resistor network can be connected between the radio device and its antenna or one could also try simply using a dummy load instead of an antenna on the APs. To avoid the hassle of physically moving client (STA), more sophisticated solution based on programmable attenuators as described in [17] can be made which automates the whole process of roaming and measurement. However, as this requires special RF equipment, we have used the built-in mechanisms in the APs and client STA to reduce the output power to a level required by this technique.

2.1.2 Active Scanning

In active scanning a STA broadcasts a probe request frame in each channel and waits for a probe response to be sent by one or more APs. The time that STA should wait for probe responses in each channel is controlled by two parameters: *min_channel_time* and *max_channel_time*. The *min_channel_time* is the time a STA must wait if it has not received any probe response or seen any traffic in that particular channel. If a STA does not get any probe response or sees no traffic in that particular channel, then that channel is assumed to be empty, and STA switches to next channel. If the STA receives any probe response or sees any traffic in that particular channel, then the STA waits in that channel until *max_channel_time* expires to receive potentially additional probe responses from other APs. Once the max_channel_time expires, then the STA processes all received probe responses and then switches to next channel.

2.1.3 Passive Scanning

In passive scanning the STA just listens for beacon frames from APs on each channel by waiting for a particular time controlled by the *channel_time* parameter. The STA processes all beacon frames from that channel and proceed sto next channel. Once all channels have been evaluated then the STA sort outs all processed beacons and selects the best AP.

2.1.4 Steps during handoff process

To simplify the handoff analysis Velayos et al. [15] proposed splitting the handoff process into three phases as shown in figure 2.2, the first phase is called the *detection phase* it starts the handoff process, actually it seems that this phase is not thus a part of handoff process although necessary for analyzing the handoff in detail and for optimizing the handoff efficiency therefore is included in handoff process analysis. The *search phase* is the second pahse during which the STA scans for candidate APs. Third phase is called the *execution-phase* in which the STA selects the appropriate AP from the scan results obtained in second phase, and tries to attach to the selected AP by sending appropriate management frames to that selected AP.

Mishra et al. [14] split the handoff process into two steps: *discovery* and *re-authentication*.

The discovery step defined by [14] is essentially the combined detection and search phase as described by Velayos et al. in [15]. While re-authentication step is the same as the execution-phase in [15].

We will follow the division of handoff process as proposed by Velayos et al. [15] because it gives greater flexibility in describing and dealing with each step of a handoff.

Detection phase

The detection phase in fact is the continues process of checking the quality of the link between STA and AP, and the triggering of the handoff process once that quality degrades to a certain pre-defined value which we call here the *handoff-threshold.* The algorithm to determine the link quality is not defined in the IEEE 802.11 standard, so it may be as simple as signal to noise ratio measurements or may combine many other parameters from the entire WLAN system. Many commercial vendors exploit this to define their own sophisticated algorithm for measuring link quality and hence perform more efficiently during handoff, as the algorithm to define link quality plays an important role in optimizing handoff latency which also reduces data loss in most scenarios.

Search phase

The search phase starts by scanning for APs. The STA must wait for *prob_delay_time* before starting scanning process, which can be either passive or active. Here we will examine the search phase when scanning is done actively.

1. Search phase starts, and after start prob delay timer,the *current_channel* is set to 0.
2. STA waits until prob delay timer reaches *prob_delay_time.*
3. STA increments *current_channel* by 1.
4. STA switches channel to *current_channel,* starts max channel timer, min channel timer , and issues probe request on *current_channel.*
5. STA listens for any probe responses and traffic on *current_channel*, until min channel timer reaches min_channel_time.
6. If no probe responses are received and the STA does not see any traffic, then the *current_channel* is assumed empty and the STA moves to step 3 to start same process for the next channel, otherwise if a probe response or traffic is seen on the *current_channel*, then STA listens on this channel until the max channel timer reaches max_channel_time.
7. STA processes all received probe responses on *current_channel* and checks if *current_channel = maximum_allowed_channel.* If not it then goes back to step 3 to start the same process for next channel.
8. Once it has gone through all channels it sorts out the processed scan results and picks the best APs (i.e., the one which may provide the best link quality).
9. Search phase ends.

Execution Phase

The execution phase utilizes the BSSID (MAC address) of best AP learned during the search phase and tries to connect to that AP as shown in figure 2.2 by sending first an authentication message to the selected AP, then once it receives a success message in an authentication response frame from this AP, it sends a re-association frame and expects a re-association response frame from this AP. This indicates that the STA is now associated with this new AP. Obviously this phase is not executed unless a suitable AP was found during the search phase.

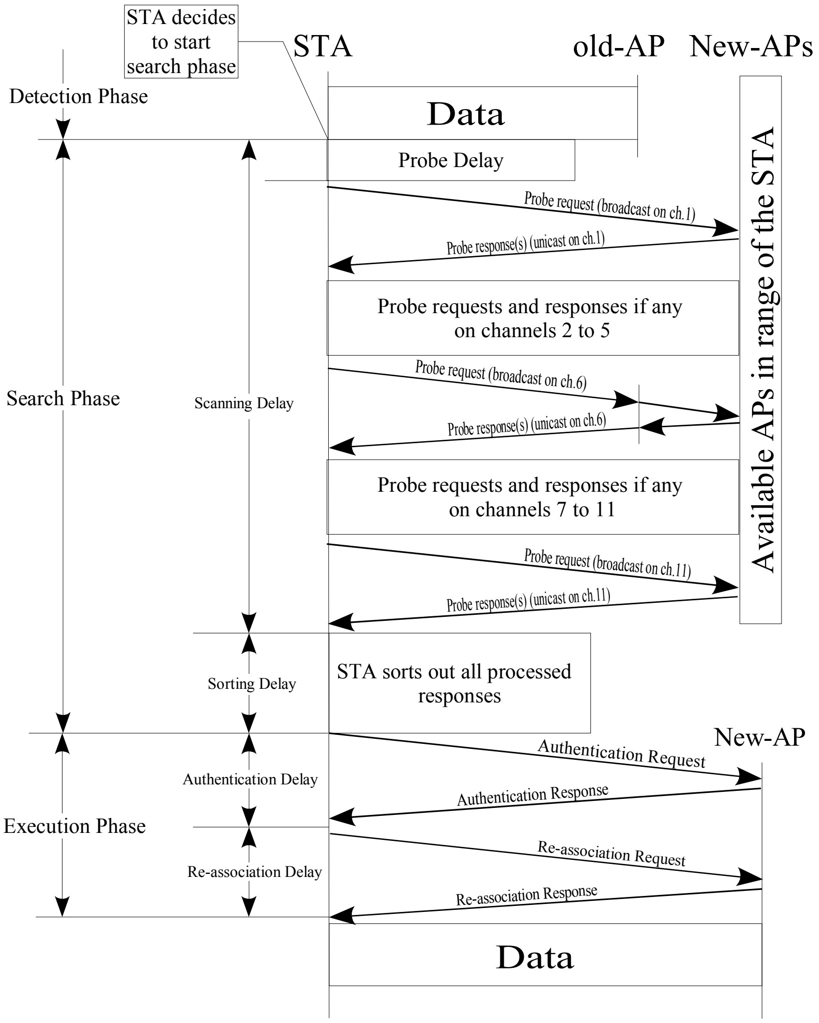

STA decides to start search phase
STA
old-AP
New-APs
Detection Phase
Data
Probe Delay
Probe request (broadcast on ch.1)
Probe response(s) (unicast on ch.1)
Probe requests and responses if any on channels 2 to 5
Search Phase
Scanning Delay
Probe request (broadcast on ch.6)
Probe response(s) (unicast on ch.6)
Probe requests and responses if any on channels 7 to 11
Probe request (broadcast on ch.11)
Probe response(s) (unicast on ch.11)
Available APs in range of the STA
Sorting Delay
STA sorts out all processed responses
New-AP
Authentication Request
Authentication Delay
Authentication Response
Execution Phase
Re-association Request
Re-association Delay
Re-association Response
Data

Figure 2.2 - Normal handoff process

Figure 2.2 shows the normal handoff process which is usually inferred from the procedures defined in IEEE 802.11 standard. From this figure we can see that the total handoff latency is the sum of delay incurred by execution phase and search phase. However, this is not the case in a real handoff as we will see later. Our tests show that different kinds of traffic patterns between the STA and AP requires that the handoff latency be measured in a different way.

2.1.5 Handoff latency

Unfortunately there is not standard way to measure handoff latency. Previous researchers [14][15][16][18] have used different criteria for measuring handoff latency. According to our understanding handoff latency can be categorized in two types: Raw handoff latency and Real handoff latency.

Raw handoff latency

We say that the raw handoff latency is the total delay incurred by the search and the execution phase of the handoff process. So thus the raw handoff latency can be measured by measuring the time interval from the first probe request by client (STA) to the re-association response from the new-AP, plus the *prob_delay_time.* So the formula for measuring raw handoff latency is,

Rhl = ProbeDelay + (ReAssResponse – FirstProbeReq)

where

Rhl = Raw handoff latency

ReAssResponse = Time stamp of Re-Association response frame from new-AP

FirstProbeReq = Time stamp of First Probe Request sent by client STA

ProbeDelay = Fixed time [section 2.1.4 – Search Phase(*prob_delay_time*)]

Also note that either there is no layer 3 traffic or it is not accounted for when measuring raw handoff latency.

Real handoff latency

However when there is layer 3 traffic in WLAN when measuring the Real handoff latency; its measurement consists of same two phases of handoff process (search phase and execution phase), but the detection phase must also be included when studying packet loss and handoff decision techniques. The most difficult phase to analyze is the detection phase because it is a nearly continuous phase; thus we must chose the correct moment in the detection phase to mark as the start of the handoff process. Real handoff can be further divided into three types: Upstream Real handoff latency, Downstream Real handoff latency, and Two-way Real handoff latency. The last being the one that we encounter in most WLAN networks.

Upstream Real handoff latency

When measuring this type of latency there is constant layer 3 traffic (usually a 172 byte UDP packet every 20ms to emulate VoIP traffic) from the STA toward host attached to the DS. When starting a handoff process, the STA usually buffers the upstream traffic until it is connected to a new-AP, therefore to measure this type of handoff latency one must account for this buffering delay. The Upstream real handoff latency can be defined as the time interval between the last UDP packet STA sent to the old-AP to being transferred to a host attached to the DS and the first packet that the STA transfers via the new-AP transferred to the same host.

Downstream Real handoff latency

When measuring this type of latency there is constant layer 3 traffic (usually 172 byte UDP packet every 20ms to emulate VoIP traffic) from a host attached to the DS to the STA which will later make a handoff. As there is only downstream traffic, the bridge(s) in the DS to which the old-AP and host are connected must learn about STA's location after successful re-association of that STA with the new-AP. To do this, a new-AP transfers a broadcast packet with STA's MAC address as the source address of the packet, in this way all bridges in the DS learns about the new location of STA so all learning bridges will subsequently transfer all downstream traffic destined for this STA through the new-AP. Therefore to measure this type of handoff delay one must take into account of the bridge(s) learning delay. The Downstream real handoff latency can be defined as the interval between the last UDP packet the STA received via the old-AP and the first UDP packet that the STA receives via the new-AP.

Bidirectional Real handoff latency

Again we assume constant layer 3 traffic (usually a 172 byte UDP packet every 20ms to emulate VoIP traffic) in both upstream and downstream directions. So both the STA and host on DS transfers UDP traffic destined to each other. The Bidirectional Real handoff latency is based on both the Upstream and Downstream Real handoff latency – the larger latency is the Bidirectional Real handoff latency.

2.2 Cross Layer Issues

Many networks, especially mobile wireless networks, need information from across layers. As we will see below this is more importantly from the lower layers to enable maximum throughput for different application usage. Thus there must be some standard defined for measurements and how to pass that information from lower to upper layers and vice versa. Even though IEEE 802 based[1] networks can take advantage of the ways defined in IEEE 802 for passing information across layers, 802.11 which is also based on 802 does not take full advantage of this standard communication technique and does not share wireless medium knowledge from physical and MAC layer with the logical link layer and upper layers. Also there is no any standard method defined for wireless link quality measurement, so different WLAN interface cards implement link quality measurement in different ways.

This cross layer communication is usually referred to as cross layer signaling becomes more important when WLAN network access is used for different purposes. For example when a WLAN STA use, the WLAN for data access, interruptions due to handoff are not usually noticeable to users. But when same STA uses the WLAN for realtime or delay sensitive communication, then even the very small interruptions caused by handoff may be noticeable, hence requiring special handling by upper layers, unfortunately the lower layers in WLAN did not give any information about these handoffs to upper layers, therefore the upper layers can't optimize accordingly. Similarly WLAN networks don't pass signal strength and noise information to upper layers, which again often results in degradation of services at the higher layers.

2.3 Previous and related work

Lots of research has been done in this area and it is still on-going. The first detailed analysis of horizontal handoff was done by Mishra et al. [14], in which they performed the handoff process and measured all the latencies at different stages of handoff. They found that:

- The specific wireless interfaces (hardware) used in the STA and the AP affects the handoff latency, and the maximum average difference in handoff latency they found was around 335.53ms when the STA was fixed and

[1] The IEEE 802 networks have same Logical Link Layer (IEEE 802.2)but can have different MAC and Physical Layer.

different AP were used, and 186.47ms when the AP was fixed and different STAs were used. This conclusion correlate with their other finding that different STAs follow different message sequences during handoff.

- The search phase accounts for around 90 percent of the total handoff latency no matter what combinations of STA and AP used.
- There are large variations in handoff latency even if one pair of STA and AP is used, the minimum of which is 75ms while maximum is around 295ms. They found out the reason is related with number of probe responses received when a probe request is sent from a STA. However it is not clearly described **why** there are large variations when same STA and AP is used in handoff.

It is important to note that they used an actual WLAN configuration for triggering the handoff (see section 2.1.1) and their handoff measurements can be categorized as raw handoff latency as they did **not** considered any layer 3 traffic during their experiments or measurements. Also they did not add the *probe_delay_time* to the total handoff latency, which is actually part of the raw handoff latency. Although they found out that different hardware (i.e., WLAN interface) had great influence on handoff latency, they did not mentioned which firmware and driver versions were used. These details are very important as using the same hardware but with different firmwares and driver, will probably produce quite different results as observed.

Velayos et al. in [15] presented techniques to reduce handoff time, and also analyzed the handoff process & presented some measurements. They concluded that:

- The detection and search phase are the main contributors to total handoff delay.
- To reduce the search phase delay they calculated optimal values for the two parameters, *min_channel_time* and *max_channel_time*, used in active scanning.
- To optimize handoff they suggest performing the detection and search phases in parallel, and provide reasoning for this suggestion.

It should be noted here that in their study they analyzed the detection phase in detail which no ealier research had done, but for handoff triggering they turn off the AP to which STA is currently attached, hence forcing the handoff. As described in section 2.1.1 this kind of handoff triggering could effect the actual handoff measurements. Further more this work also did not specify the driver and firmware versions. They mention that the STA generated a flow of packets with the “characteristics of voice over IP”, but do not give more details about this traffic (such was it upstream or downstream or with what packet interval or in fact information about generated traffic). Probably because of this they also assume that some solutions are used which cause the upstream and downstream handoffs equal.

J-O Vatn [16] studied handoff performance and its effects on voice traffic, so the main difference from [14] is that there is a voice like traffic between STA and a host connected to the DS. This research resulted in following findings:

- Just as in [14], the search phase is the longest of all phases in handoff and its length depends on the STA in use.
- The STA may receive data from the old-AP during search phase.
- The exact handover behavior not only depends on type of hardware(WLAN interface) used in STA and AP, but also on the data flow (i.e., direction) between STA and AP.
- The power save trick used by the D-Link (prism 2/2.5/3 chipset based) STA to enter sleep mode before starting search phase and entry into wakeup mode after finishing search phase is used to increase buffering by the AP.
- During the handover upstream data packets are usually delayed while downstream packets are most often lost.

Worth noting in this research is that there was VoIP like layer 3 traffic (UDP packets of 172 byte) between the STA and AP during handoff; hence we can say that the measurements correspond to the handoff latency categorized as Real handoff latency. This study also talks about and examines the difference in handoff latency when there is only upstream or downstream traffic between the STA and AP. The handoff was triggered by decreasing the old-APs output power abruptly to force handoff, while STA was stationary (see section 2.1.1). This triggering might have some effect on results as an abrupt decrease in signal strength may have effects on the detection phase of the handoff, which is included when measuring Real handoff latency. Apart from the delay measurements of different phases of handoff this research also presents details of packet loss and packet delays incurred during handoff process. The other good thing about this research is that all the version numbers of both firmware and driver are explicitly mentioned which should help other researchers when the same hardware is used.

Shin et al. in [18], uses selective scanning to reduce the search phase, and also use a neighbor caching algorithm at the STA to avoid the search phase, hence greatly reducing the overall handoff latency in open WLAN networks. They also use VoIP traffic in their tests and present packet loss during handoff. Their suggested solution only requires changes at STA side. However, WLAN networks which are secured using 802.11i wont benefit from their proposed solution and this solution may not work well when the APs are concentrated, heavily overlapped and use more than 3 to 5 channels in the WLAN system. As with many other researchers they have not described the version numbers of the driver and firmware software which they used.

In addition to the above there have been many studies in this area of which [19] and [20] are focused on reducing 802.1X authentication time, but require lots of changes in the AP, STA and the Authentication Server. Kim et el. [21] also uses a selective scanning which is little bit different from [18] and introduces a dedicated server to provide data for selective scanning, as well as requires that the AP and STA, and that server communicate through some protocol to exchange information.

It is Important to note that the previous work mentioned above focus on for optimizing handoffs in **open** WLANs and do not talk or evaluate their solutions when the WLAN is secured according to the IEEE 802.11i standard.

3 Methodology

Although we currently are able to optimize handoff latency using techniques and suggestions folowing [18][19][20][21], as described in section section 2.3 some changes both the AP as well as the STA; additionally none of the previous studies addressed 802.11i enabled WLANs. Also in some scenarios when APs are concentrated the solutions don't work efficiently.Thus the goal of this thesis was to propose handoff optimization techniques for WLANs when full 802.11i security is in place and that these techniques should works in all scenarios. This means that we must begin by analyzing and optimizing current algorithms for reducing handoff latency and adopting suitable ones for 802.11i enabled WLANs.

To achieve this, we first have to design a testbed where the handoff process can be reproduceably performed and analyzed.

3.1 The Testbed

The design of testbed is shown in figure 3.1. It includes at least two APs, one client STA, a Hub acting as the DS, and one PC with two WLAN interfaces to monitor the traffic on two channels.

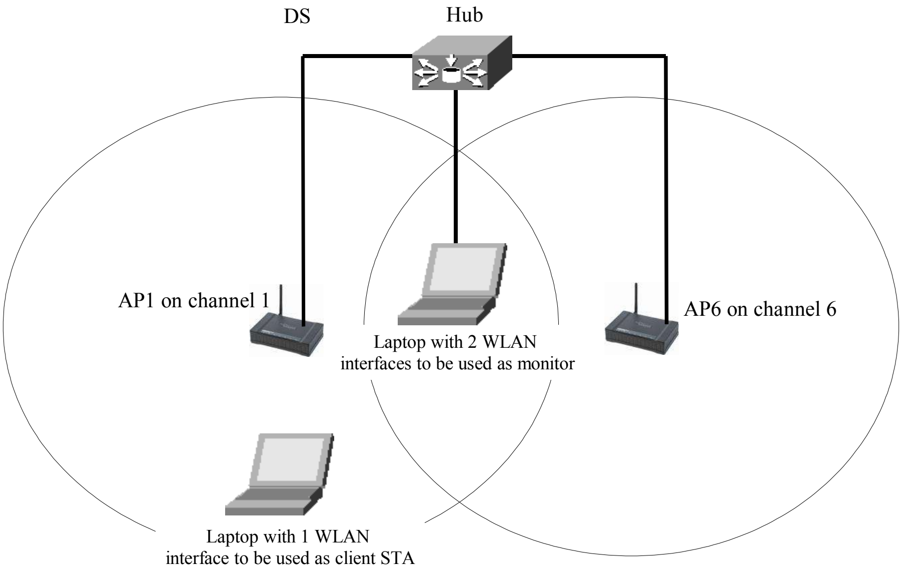

DS
Hub
AP1 on channel 1
Laptop with 2 WLAN interfaces to be used as monitor
AP6 on channel 6
Laptop with 1 WLAN interface to be used as client STA

Figure 3.1 - General testbed design

The selection of hardware for the testbed has to met many requirements due to the nature of the experiments and the flexibility needed to controll the WLAN interface. The main factors considered when selecting the hardware included:

- The WLAN interfaces must allow control of all three phases of handoff (see section 2.1.4)
- The WLAN interface both in AP and STA must support full 802.11i, including pre-authentication.

- The WLAN driver should be readily available for a Linux platform.
- At least two WLAN interfaces must be able to operate in monitor mode to capture the WLAN traffic and also possibly capture the (802.11 radio) header of frames.
- The WLAN interface should allow operation as either AP or STA.
- The WLAN interface in the AP should also be as flexible as in STA.

Considering these requirements and surveying WLAN market, we learned that there was no AP or client STA on the market in Dec 2004 which has support from the hardware vendor for full 802.11i with PMKSA caching and 802.11i pre-authentication as defined in IEEE 802.11i[8] standard. During our survey we also learned about an open source WLAN driver called HostAP[22], which is vendor independent and supports any PCMCIA/PCI WLAN card based on the prism2/2.5/3 chipset. Furthermore it seemed to be consistent with most of our requirements **including** full conformance with 802.11i.

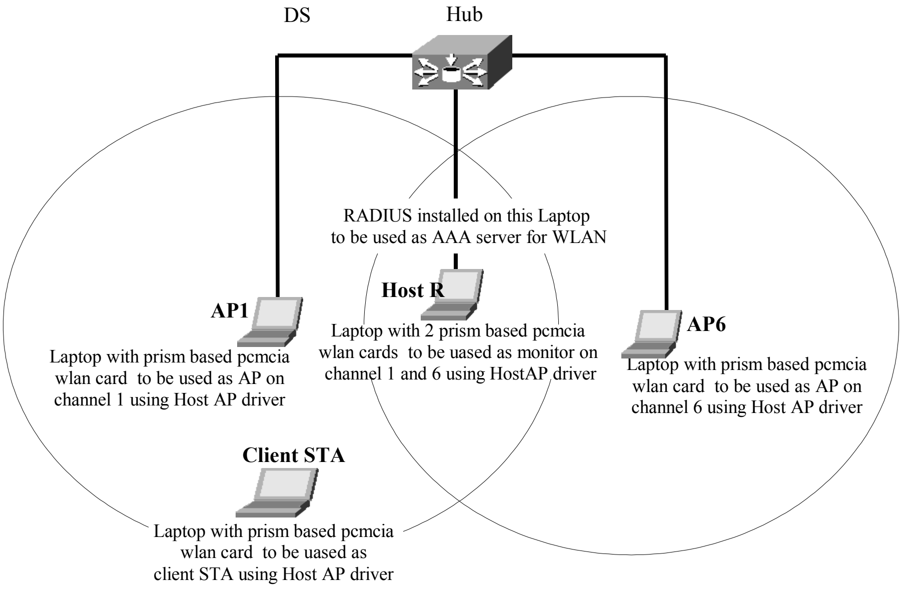

DS
Hub
RADIUS installed on this Laptop
to be used as AAA server for WLAN
Host R
Laptop with 2 prism based pcmcia
wlan cards to be uased as monitor on
channel 1 and 6 using HostAP driver
AP1
Laptop with prism based pcmcia
wlan card to be used as AP on
channel 1 using Host AP driver
AP6
Laptop with prism based pcmcia
wlan card to be used as AP on
channel 6 using Host AP driver
Client STA
Laptop with prism based pcmcia
wlan card to be uased as
client STA using Host AP driver

Figure 3.2 - The testbed used for all experiments

Thus we decided to use Laptops with PCMCIA slots both for APs, client STAs, and monitoring . Several PII laptops and the other required hardware was provided by R^{2}M. The full testbed is shown in figure 3.2. To make the testbed 802.11i compliant, RADIUS (free RADIUS [23]) was used as the AAA server. The main reason for using RADIUS instead of DIAMETER was that the DIAMETER protocol is still in development hence there were not full implementations available. APs were made using the hostapd part of the HostAP driver, while each STA uses the HostAP driver. The 802.11i functionality in APs and STAs can be turned on or off. Configurations of all nodes are given in appendix B.

3.2 HostAP Driver and Prism 2/2.5/3 chipset

HostAP[22] is an open source driver suite for PCMCIA and PCI WLAN cards based on the prism 2/2.5/3 chipset. The hostapd user space program of this suite used to make these WLAN cards work as an AP, hence it provides the management functions which a typical AP does. It also fully supports all the same 802.11i functions which a typical 80.2.11i enabled AP does. There is a developers manual [24] for the prism 2/2.5/3 chipset, which we used to access internal registers for direct control of some functions which the HostAP driver does not provide. On the STA side we used the HostAP driver and a WPA supplicant from the HostAP driver suite which provides full 802.11i support for STA WLAN clients.

We tried different versions firmware and found out that version 1.7.4 was the best in all respects, thus we flashed our WLAN cards with that firmware, and we used 0.3.7 version of the HostAP driver, hostapd and WPA supplicant. We used two ASUS SpaceLink 802.11b WLAN card (WL-100)for the APs and one Zyxel ZyAir B-101 (802.11b) as STA. We also used one Zyxel ZyAir B-101 (802.11b) as a monitor on one channel and Netgear WLAN card as a monitor on the other channel.

3.3 VoIP traffic generation

We used MGEN(Multi-GENerator)[25] to generate 172 byte UDP packets every 20 ms to emulate a 64 kbit/s pulse code modulated (PCM) voice stream packetized into 160 byte units plus a 12 byte RTP (Real Time Protocol) header.

3.4 Traffic monitoring and measurement

We used ethereal to capture the WLAN frames on two channels using two WLAN cards in monitor mode. During some experiments we also turned on the radio header capture feature in WLAN cards, thus we were able to see the 802.11 as well as radio headers of all the WLAN frames.

3.5 Handoff Trigger

We triggered handoff as explained in last paragraph of section 2.1.1 by reducing power of both APs to such an extent that they cover just an small area in a small room, so that a STA will make a handoff by simply moving the STA between these two APs. We also de-attached antenna of WLAN card in STA.

3.6 Experimental Setup

Before starting the full tests, we performed some preliminary tests and measurements, to understand the WLAN environment and learn exactly how everything worked. We found out that there was a 20ms deviation in the measurements of same kind, so we decided to do 20 repetitions of every test, in order to give an average value with little variance.

As the layer-3 traffic source only generated a packet every 20ms we chose to only measure packet delay due to handoff , hence only 20 test runs were needed for each test to have a measurement within source packet variance.

4 Tests, Measurements and Observations

This section presents experiments and measurements performed in different scenarios to analyze in detail the handoff process in different situations.

4.1 Handoff process – No Layer 3 Traffic

In this section we present raw handoff tests and measurements i.e., when there no layer 3 traffic between AP and STA. These tests were performed to analyze how the firmware of the WLAN card performs the handoff and to compare it with the expected handoff process. Section 4.1.1 presents a raw handoff process when the security features are not enabled while section 4.1.2 presents the raw handoff process when the WLAN network is secured utilizing the IEEE 802.11i standard.

4.1.1 No Security features enabled – No Layer 3 Traffic

In this test the WLAN network in the testbed has has no security and there is no Layer 3 traffic. That means 802.11i security features in the testbed have been disabled and traffic generation was turned off for this test. The goal was to identify tiny details and parameters which can be used and tuned in later tests. The configuration of all nodes are in appendix B.

As shown in figure 4.1, the raw handoff process observed during our tests is little bit different than shown in figure 2.2. The difference being the Null Function Data frame(s) observed before and after search operation. Usually Null Function Data frame are sent by the STA for Power Save (PS) mode, however here they are not sent for power saving but rather for buffering. The Null Function Data frame before scanning is sent by the STA to tell the old-AP that this STA is going to sleep before scanning and the later Null Function Data frame tells the old-AP that the STA has now awakened. Through this trick the AP can buffer any packets arriving for this STA during scanning. When the scanning finishes the old-AP will try to deliver it, of course for this STA still must be within the range of the old-AP.

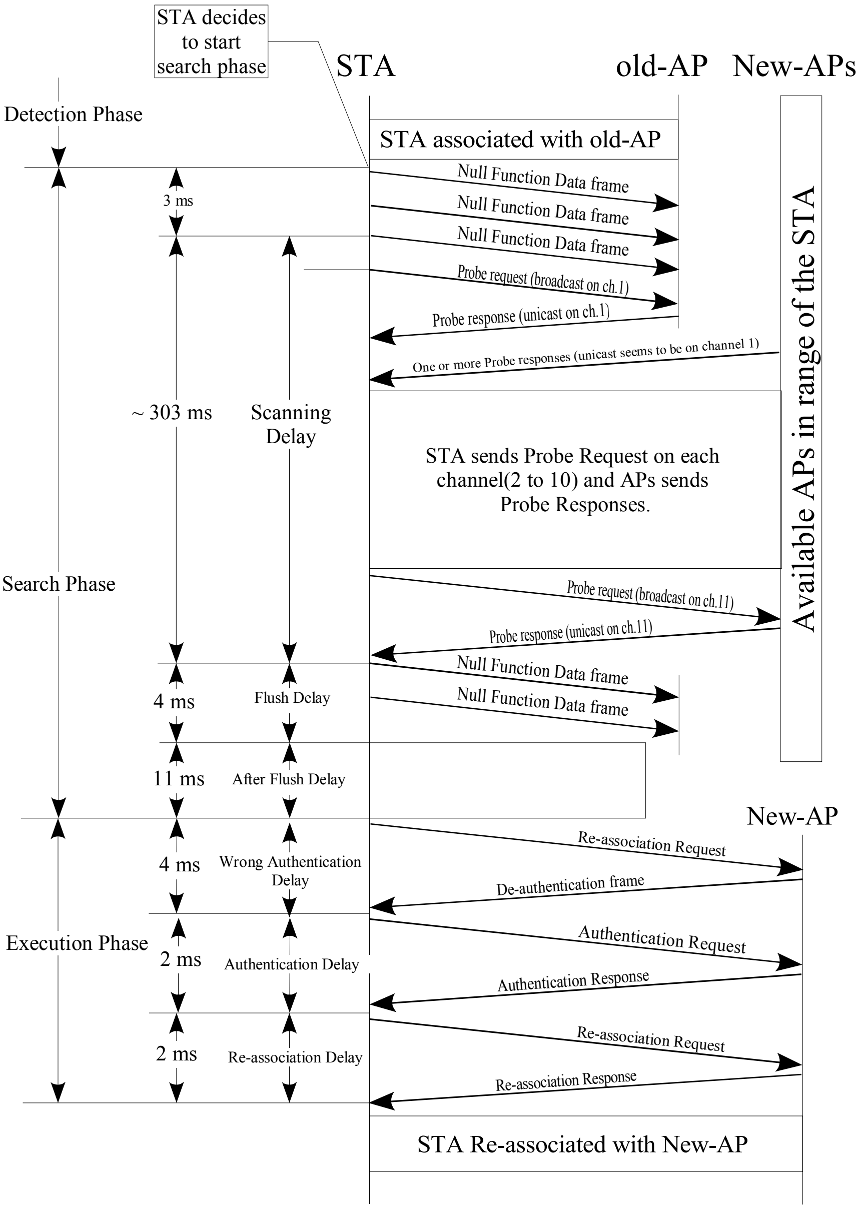


Figure 4.1 - Handoff process observed in our testbed – No L3 traffic and No security

A total of 20 test runs were made and the average observed timings for different phases during handoff are in table 4.1.

We see here that before scanning STA sends three Null Function Data frames, on inspection we found out that only the last two of the three are retransmitted as the STA did not get any ACKs from the old-AP. However re-transmitting non-ACKed frames is usual procedure for any frame transferred from STA to AP.

Handoff Phases	**Mean (ms)**
Buffer Delay (Sleep Trick)	3.3
Scanning Delay	303.4
Flush Delay (Wakeup Trick)	4.1
After Flush Delay	10.9
Total Search Phase Delay	*321.7*
Wrong Authentication Delay	4.2
Correct Authentication Delay	2.1
Re-association Delay	2.3
Total Execution Phase Delay	*8.6*
Total Handoff Delay	**330.3**

Table 4.1 - Raw handoff timings

4.1.2 802.11i enabled WLANs – No Layer 3 Traffic

This test is similar to test described in section 4.1.1 above, the only difference being that 802.11i security features have been enabled, by introducing the 802.1x security model and RADIUS server as defined in section 1.4. Here there is layer 3 traffic, but it is only on DS i.e., on wired network between the RADIUS server and the APs. The RADIUS server was installed on Host R, as shown in figure 3.2. SSL certificates were used for authentication. The HostAP driver on AP1, AP6, and STA was configured to use 802.11i. A total of 40 test runs were made out of which 20 were **without** pre-authentication(as defined in 802.11i) enabled, while the remaining 20 were **with** pre-authentication enabled. Also note here that AP1 and AP6 were placed close to each other in such a way so that STA finds both APs during scanning hence it knows the RSN IE of new AP before doing pre-authentication, which is the requirement for pre-authentication in IEEE 802.11i (as described in section 1.5.6). Again all the node configurations in this experiment are described in appendix B.

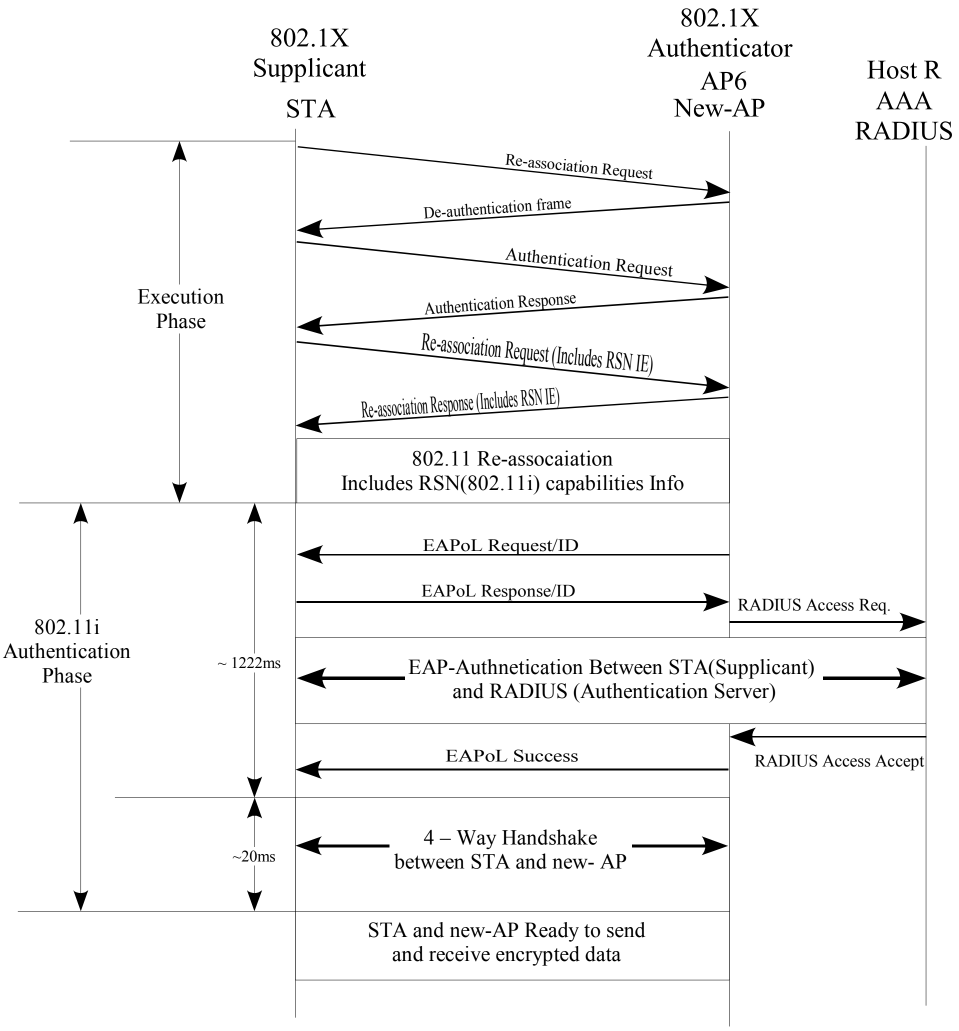


Figure 4.2 - Message flow during and after execution phase in handoff when WLAN in testbed secured using 802.11i

Here in table 4.2 we can see that when the WLAN is secured using 802.11i, the

Handoff Phases	**Mean (ms)**
Total Search Phase Delay	*330.3*
Wrong Authentication Delay	4.5
Correct Authentication Delay	1.9
Re-association Delay	2.1
Total Execution Phase Delay	*8.5*
EAP Authentication Delay	1222.4
4 way handshake Delay	20.2
Total 802.11i Delay	*1244.6*
Total Handoff Delay	**1583.4**

Table 4.2 - Raw Handoff timings in 802.11i secured WLAN networks

handoff latency is increased and the total handoff delay is approximately 5 times the original delay, when the WLAN network was not secured.

Table 4.3 shows the timings of the pre-authentication process which is done through the old-AP as described in section 1.5.6, while table 4.4 shows the handoff timings when the STA does a handoff to the new-AP, to which it already has pre-authenticated through the old-AP.

Pre-Authenticated Handoff Process	**Mean (ms)**
EAP Pre-Authentication Delay (through old-AP)	1245.4
Total Pre-authentication process Delay (at old-AP)	*1245.4*

Table 4.3 - Pre-Authentication Process timings at old-AP

Pre-Authneticated Handoff Phases	**Mean (ms)**
Total Search Phase Delay	*317.3*
Total Execution Phase Delay	*7*
Total 802.11i latency at new-AP (4 way handshake Delay)	*20.1*
Total Handoff Delay	**344.4**

Table 4.4 - Handoff timings of Pre-authenticated STA at new-AP

Here we observe that when pre-authentication is used, the 802.11i latency reduces to just the time required for the 4-way handshake, this is 20ms.

We also observe that the latency introduced by 802.11i is nearly constant. The 802.11i authentication starts only after a STA has already associated with the new-AP hence the delay variance in 802.11i authentication latency would only be because of delay in the DS, that is the communication between the new-AP and the RADIUS Authentication Server. So we don't need to do extensive further tests with an 802.11i enabled WLAN to learn overall handoff latency, as we just need to add 802.11i latency from this test to the tests which were done without 802.11i secured networks, to find the overall latency.

4.2 Handoff process – with unidirectional VoIP like Traffic

In this section handoff process occurs while there is only unidirectional VoIP like traffic in the WLAN system, that is either from the STA to DS or vice versa. Most of the time during a voice session between two or more persons only one person speaks at a time, thus many VoIP end clients take advantage of this by using silence detection[1] which makes it possible to transfer VoIP packets in one direction during that time without degradation in quality or knowledge of user. Here a one-way traffic stream is used to simulate such silence detection. The MGEN traffic generator was used to generate one way VoIP like traffic as described in section 3.1.2. The goal is to identify packet loss and latency for VoIP during the handoff process, and observe any effects

[1] In silence detection, each VoIP clients detects if the user is speaking or not, then it does not send any voice.

due to VoIP traffic on the handoff process or the handoff latency.

4.2.1 Upstream traffic only

In this test there is only upstream traffic from STA to AP, thus in our testbed the STA sends VoIP packets to Host R. Figure 4.3 shows the message flow during the handoff process when there is only upstream traffic from STA to AP.

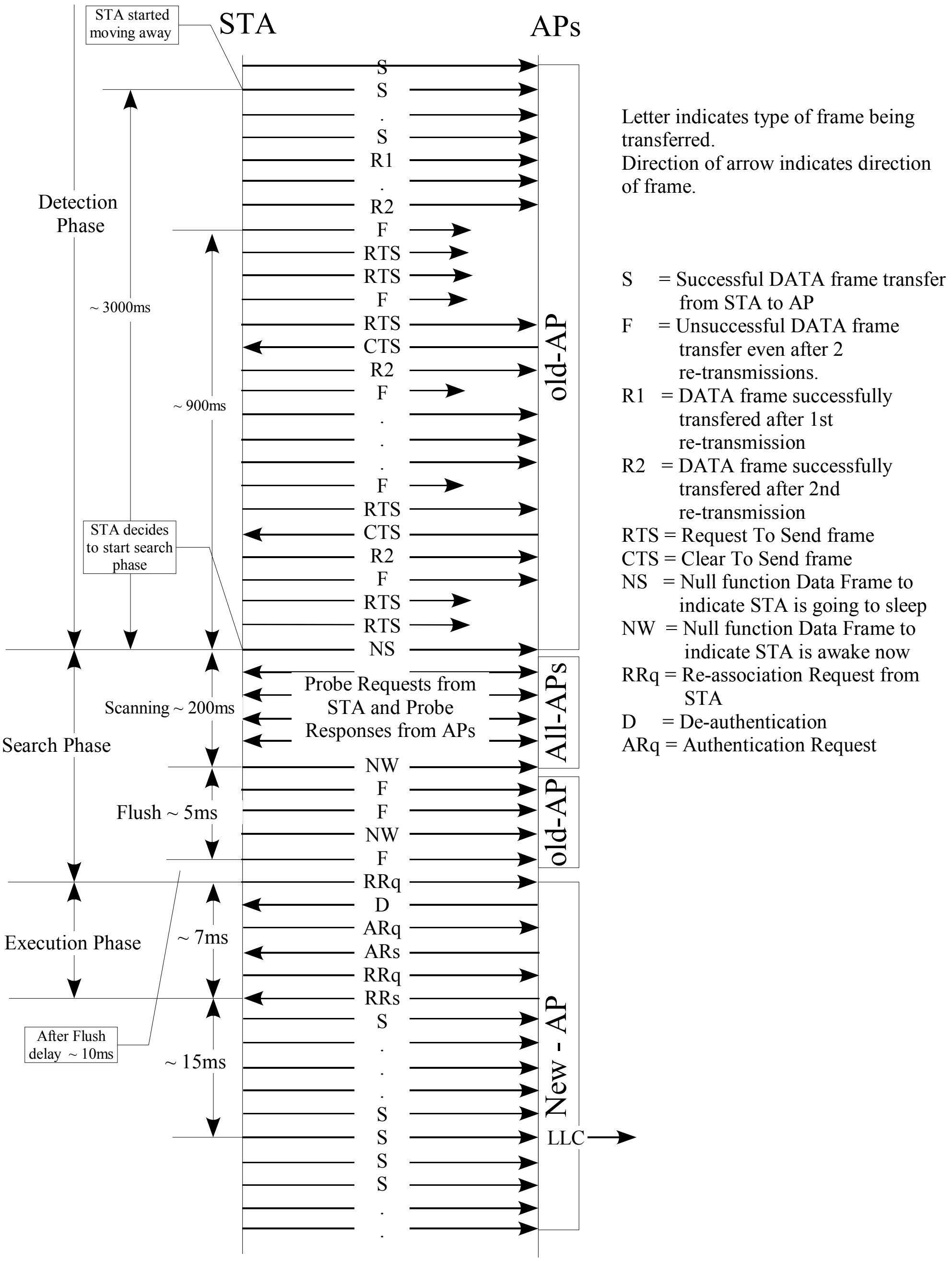

STA started moving away
STA
APs
S
S
.
S
R1
.
R2
F
RTS
RTS
F
RTS
CTS
R2
F
.
.
.
F
RTS
CTS
R2
F
RTS
RTS
NS
Probe Requests from STA and Probe Responses from APs
NW
F
F
NW
F
RRq
D
ARq
ARs
RRq
RRs
S
.
.
.
S
S
S
S
.
.
LLC
old-AP
All-APs
old-AP
New - AP
Detection Phase
~ 3000ms
~ 900ms
STA decides to start search phase
Search Phase
Scanning ~ 200ms
Flush ~ 5ms
Execution Phase
~ 7ms
~ 15ms
After Flush delay ~ 10ms
Letter indicates type of frame being transferred.
Direction of arrow indicates direction of frame.
S = Successful DATA frame transfer from STA to AP
F = Unsuccessful DATA frame transfer even after 2 re-transmissions.
R1 = DATA frame successfully transfered after 1st re-transmission
R2 = DATA frame successfully transfered after 2nd re-transmission
RTS = Request To Send frame
CTS = Clear To Send frame
NS = Null function Data Frame to indicate STA is going to sleep
NW = Null function Data Frame to indicate STA is awake now
RRq = Re-association Request from STA
D = De-authentication
ARq = Authentication Request

Figure 4.3 - Handoff process with Upstream traffic

Average of timings from 20 test runs are given below in table 4.5

Handoff Phases	**Mean (ms)**
Buffer Delay (PS Trick - Sleep)	1
Scanning Delay	200.3
Flush Delay (PS Trick - Wakeup)	5.3
After Flush Delay	10.4
Total Search Phase Delay	*217*
Wrong Authentication Delay	3.2
Correct Authentication Delay	2.1
Re-association Delay	2.3
Total Execution Phase Delay	*7.6*
Total Handoff Delay	**224.6**

Table 4.5 - Handoff timings when there is only upstream traffic towards STA

Here we observe that the scanning delay is reduced considerably as compared to the earlier raw handoff timings, but there is little difference in other phases of handoff.

	Detection Phase	**Search Phase**	**Execution Phase**	**Total**
Packets Lost	33	3	0	36
Packets Delayed	0	11	1	12

Table 4.6 - Number of packets lost and delayed because of handoff

Table 4.6 shows us the number of packets lost and delayed during different phases of the handoff. We observe here that the most of the packets are lost during the detection phase or more specifically just before the search phase. Although STA tries to send some buffered packets during the flush phase [16] they are lost. Successful transmission of packets starts once the execution phase finishes and the STA has successfully associated with the new-AP. So the first successful packet delivered through the new-AP is delayed around 194ms which is the total handoff delay minus 20ms for every packet lost during search and execution phase. Also note that link layer update packet is transferred by the new-AP15ms after the association response from the new-AP to the STA. However, this does not have any significant role here as only upstream traffic is transferred, but we will see its effect when there is only down stream traffic during handoff.

4.2.2 Downstream traffic only

In this test there is only downstream traffic from AP to STA, so in our testbed Host R sends VoIP packets to STA. Figure 4.4 shows the message flow of the handoff process when there is only downstream traffic.

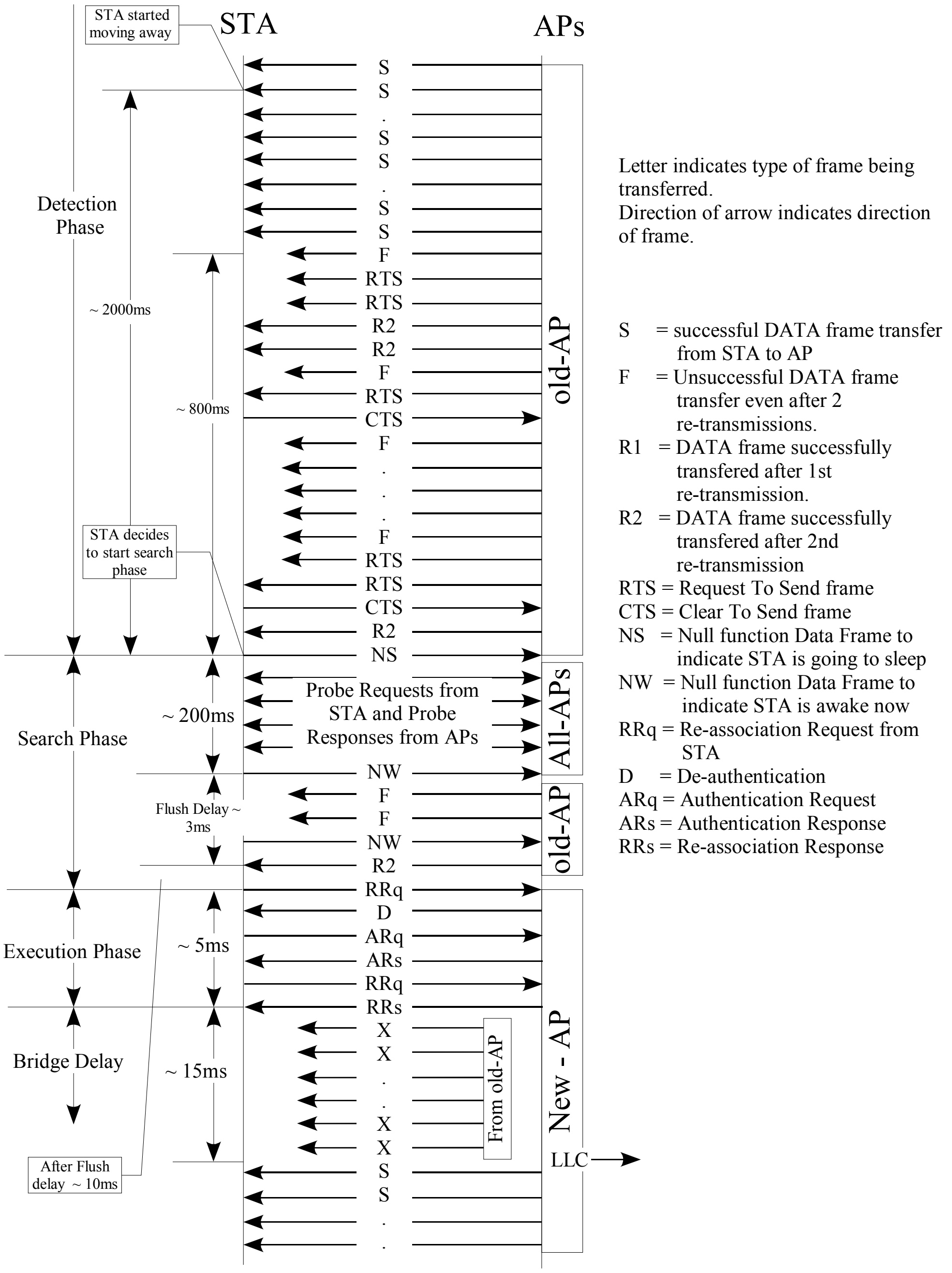

STA started moving away
STA
APs
S
S
S
S
S
S
F
RTS
RTS
R2
R2
F
RTS
CTS
F
F
RTS
RTS
CTS
R2
NS
Probe Requests from STA and Probe Responses from APs
NW
F
F
NW
R2
RRq
D
ARq
ARs
RRq
RRs
X
X
X
X
S
S
Detection Phase
~ 2000ms
~ 800ms
STA decides to start search phase
Search Phase
~ 200ms
Flush Delay ~ 3ms
Execution Phase
~ 5ms
Bridge Delay
~ 15ms
After Flush delay ~ 10ms
old-AP
All-APs
old-AP
New - AP
From old-AP
LLC
Letter indicates type of frame being transferred.
Direction of arrow indicates direction of frame.
S = successful DATA frame transfer from STA to AP
F = Unsuccessful DATA frame transfer even after 2 re-transmissions.
R1 = DATA frame successfully transfered after 1st re-transmission.
R2 = DATA frame successfully transfered after 2nd re-transmission
RTS = Request To Send frame
CTS = Clear To Send frame
NS = Null function Data Frame to indicate STA is going to sleep
NW = Null function Data Frame to indicate STA is awake now
RRq = Re-association Request from STA
D = De-authentication
ARq = Authentication Request
ARs = Authentication Response
RRs = Re-association Response

Figure 4.4 - Handoff process with only Downstream traffic toward STA

From above figure we observe that downstream handoff process is quite similar to the upstream handoff process, but with the slight difference that the first successful packet delivered to the STA after the execution phase is after 15ms after the end of the execution phase. The reason is the link layer update packet, which notifies all nodes on the DS of the new location of the STA. The new-AP sent this 15ms after sending the association response frame to STA. For upstream traffic this was not a problem as the DS learned about the new location of the STA from the first packet that the STA transferred through the new-AP, but in this test as there is no upstream traffic, thus the first successful transmission of packet after the execution phase depends on the link later update packet being sent by the new-AP to the DS.

Table 4.7 gives average handoff delays during different phases of handoff in this test. Observe the 15.3 ms bridge delay is included in the total handoff timings, as explained in the above paragraph.

Handoff Phases	**Mean (ms)**
Buffer Delay (PS Trick - Sleep)	1
Scanning Delay	197.3
Flush Delay (PS Trick - Wakeup)	3.2
After Flush Delay	10.4
Total Search Phase Delay	*211.9*
Wrong Authentication Delay	3.3
Correct Authentication Delay	2.5
Re-association Delay	2.3
Total Execution Phase Delay	*8.1*
Total LLU Delay	*15.3*
Total Handoff Delay	**236.3**

Table 4.7 - Handoff timings when there is only downstream traffic towards STA

	Detection Phase	**Search Phase**	**Execution Phase**	**Total**
Packets Lost	33	11	1	44
Packets Delayed	0	0	0	0

Table 4.8 - Number of packets lost and delayed because of handoff

Table 4.8 gives us the number of packets lost and delayed during the different phases of this handoff, we observe here that as in previous test the majority of packets were lost during the detection phase specifically just before the search phase. The main difference from the previous case is that almost all packets are lost here and none were delayed; the reason is that packets which were buffered during search the phase, are lost since the STA can no longer receive all packets from the old-AP, even though it may receive some during flush phase, but we observed that it did not receive even one packet during the flush phase. Further more until the DS receives some update about STA's new location, it continues to send packets every 20ms to the old-AP which are subsequently lost. Only after the DS has received the LLU frame from the new-AP, then it begins to forward packets to the new-AP.

4.3 Handoff process with bi-directional VoIP like traffic

In this section handoff is examined when there is bi-directional VoIP like traffic in the WLAN system, that is there is traffic from STA to AP **and** from AP to STA at the same time during the handoff. In our testbed, VoIP like packets are being sent from the STA to Host R and from Host R to the STA during the handoff. MGEN[25] traffic generator was used at both ends to generate VoIP like traffic as described in section 3.3. The goal is to identify packet loss and latency for VoIP traffic during the handoff process, and to observe any effects of VoIP like traffic on the handoff process.

Table 4.9 gives the average handoff delays during the different phases of handoff obtained from 20 test runs of this type of handoff. We see that these are in line with the earlier tests involving unidirectional traffic. Observe here that LLU packet is transferred in after-execution phase after 20ms, but unlike in downstream-unidirectional traffic case, here it only effects either zero or one downstream packet, the reason is that first upstream packet from STA updates the location of the STA in the DS after which downstream packets are delivered to the correct AP.

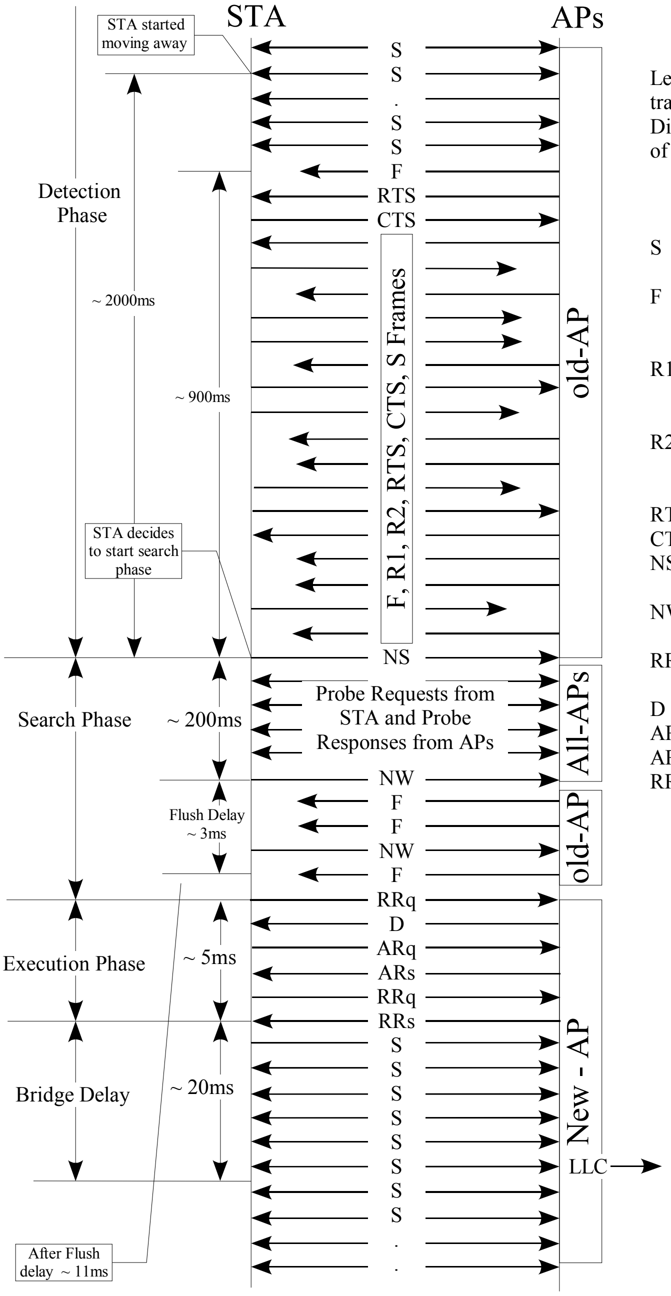
Handoff Phases	**Mean (ms)**
Buffer Delay (PS Trick - Sleep)	1
Scanning Delay	202.3
Flush Delay (PS Trick - Wakeup)	3.2
After Flush Delay	11.1
Total Search Phase Delay	*217.6*
Wrong Authentication Delay	3.1
Correct Authentication Delay	2.5
Re-association Delay	2.3
Total Execution Phase Delay	*7.9*
Total LLU Delay	*1.2*
Total Handoff Delay	**226.7**

Table 4.9 - Handoff timings when there is bi-directional traffic

Table 4.10 gives us the number of packets lost and delayed during the different phases of this handoff. These measurements are also in-line with previous upstream and downstream cases, which showed that most packets were lost in the detection phase. The only difference is that one packet was **not** sent to the wrong AP (old-AP) after re-association since the upstream packet updated the location of the STA to the DS. Figure 4.5 shows the message flow during this test.

Packets	Detection Phase	Search Phase	Execution Phase	Total
Lost (Upstream)	34	3	0	37
Delayed (Upstream)	0	11	1	12
Lost (Downstream)	35	10	1	46
Delayed (Downstream)	0	0	0	0

Table 4.10 - Number of packets lost and delayed because of handoff



Letter indicates type of frame being transferred.
Direction of arrow indicates direction of frame.

S	= successful DATA frame transfer from STA to AP
F	= Unsuccessful DATA frame transfer even after 2 re-transmissions.
R1	= DATA frame successfully transfered after 1st re-transmission.
R2	= DATA frame successfully transfered after 2nd re-transmission
RTS	= Request To Send frame
CTS	= Clear To Send frame
NS	= Null function Data Frame to indicate STA is going to sleep
NW	= Null function Data Frame to indicate STA is awake now
RRq	= Re-association Request from STA
D	= De-authentication
ARq	= Authentication Request
ARs	= Authentication Response
RRs	= Re-association Response associated to new-AP.

Figure 4.5 - Handoff process with bi-directional traffic

4.4 Handoff Process - More Experiments

As we want to extract some parameters from search phase we conducted additional experiments on the search phase by utilizing some scanning options and parameters provided by HostAP driver and by directly manipulating the contents of internal registers as described in Prism developer's manual [24]

1. Probe Responses from all APs in range are received within 1ms to 3ms of the Probe Request from the STA.
2. The values of three parameters:

   probe_delay_time = 10ms

   max_channel_time = 60ms

   min_channel_time = 15ms

   are fixed in the firmware and we could not find anyway to change them.
3. Interestingly most of the time during scanning the STA does not scans all channels! Other researchers in this area have observed this before; probably because this requires either a monitoring station on every channel or some technique for inspection of received frames is needed. We used sequence numbers in the the 802.11 headers to observe this. Later we confirmed this when sometimes we could not find our AP during scanning in these experiments.

1. The probe request sent on a particular channel leads to probe responses not only from AP on that particular channel, but also from APs on neighboring channels. This leads to the STA (specifically STA's firmware) believing that there is a traffic or an AP in that channel hence it waits until max_channel_timer for to scan this channel. But it is not clear if the STA's firmware delivers these responses to the driver or not, and even if it delivers then, it is unclear whether the driver uses that response in the search results or not.

4. There is no explanation why the STA waits for around 10ms in the “after-flush” phase.

5 Analysis and improvements

5.1 Analysis – Search Phase

From the handoff tests we observe that the search phase latency difference is around 100ms between raw handoff and real handoff, to find out the reason we did more experiments and results and observations from those experiments are in section 4.4. We found out that the STA does not necessarily scan all channels during scanning, and we believe this is one of the reason for large difference in search latency between raw handoff and real handoff, as the number of channels scanned greatly influence the total search phase latency. The other reason is the undesired Probe Responses from APs, see section 4.4. which unnecessarily increases the the STA's wait time on a particular channel to *max_channel_time,* and this time is roughly 4 times the *min_channel_time* which STA would have waited if it had **not** received any Probe Responses from APs. We also believe that these two reasons are responsible for the large search phase latency variations between same type of handoff experiments by other researchers – even if the handoff tests were done with exactly the same settings as [14]. For the same reasons that Vatn [16] observed large standard deviations in the search phase latency for D-Link WLAN card which is the same prism chipset as WLAN cards in our experiments.

Furthermore in the search phase we observe that none of the packets get through during flush period, probably because the STA was too far away from the the old-AP to send or receive packets through the old-AP. The flush interval was only mentioned in [16], where sometimes one packet gets through, but the handoff triggering mechanism used in that study is different one than used in this one. Because of that most of the time the STA in his experiments could communicate with the old-AP even during search the phase, and hence could receive and send some packets during flush period.

5.2 Improvements – Search Phase

As we see from our experiments the major part of the search phase latency is composed of scanning delay, thus a very simple, easily adoptable, and fast solution to decrease the scanning delay would be to optimize the values for *probe_delay_time, min_channel_time and max_channel_time. Probe_delay_time* simply can be reduce to 1 to 2ms. As we have observed that probe responses from APs are received within 1 to 3ms of the probe request, and Velayos et el. [15] found out that even if the AP is loaded with 10 STAs then 10ms would be the optimal *max_channel_time* value. So we suggest 3ms for *min_channel_time* and 10ms for *max_channel_time.* The key to understanding these values is that if there is lot of traffic on that channel then the STA would notice that within 3ms and hence wait until 10ms, and probably with lots of traffic AP may be loaded so it may not answer within *min_channel_time*, but would answer within *max_channel_time*, on the other hand if there is no traffic on that channel then probably either there is no AP on that channel or if there is an AP then other STAs are attached but are not transferring/receiving at that particular time. In the former case the STA would not wait more than 3ms in that particular channel which is what we want, and in the later case as AP is not loaded (not transferring/receiving at that particular time) so it would answer within 3ms of the probe request. With above suggestions we would be able to reduce scanning latency from 330ms to 60ms in an ideal environment where there is a traffic on 3 channels out of 13 channels. This solution is very flexible, it requires only very small changes on the STA side and hence could work in current WLAN deployments without any changes to the AP.

We can further reduce this scanning delay with some more dynamic changes which are little bit more complex than the previous ones, but again only on the STA side, hence they would work on existing WLAN infrastructure. This requires *max_channel_time* to be a dynamic variable so that its value can be changed dynamically based on the received responses. This would require processing the received responses on a per response basis or a per-channel basis, but the later is easy to implement and does not require extensive changes or extensive communication across either physical or MAC layer, while the former requires dynamic cross layer communication during the whole scanning interval. The main idea behind processing received responses on a per channel basis is that after every scanned channel, the received probe responses are processed and then a decision is made to continue scanning or not, and if it is decided to continue scanning then the *max_channel_time* for the next channel (s) could be set accordingly. These decisions are based on the criteria that if the processed probe response(s) are above the required signal threshhold or not. The same idea could be used when processing received probe responses on a per response basis to decide if it should continuing scanning and to set *max_channel_time* after receiving every probe response. The key concept is that if a STA has already found a good (signal above threshhold) AP, then there is no need to scan further, just re-associate with that one. With this technique the scanning latency can be as low as 5ms in the best case where a good probe response is received within *min_channel_time* of first probe request, and 55ms in worst case where a good probe response is received on last scanned channel. This could be improved if an heuristic algorithm decides the order of scanning (i.e., which channel to scan first, second .... last), where the probability of receiving a good response on a particular channel depends on how early is that channel is in the scanning sequence. With this technique the probability is higher that the scanning latency would be closer to 5ms in most cases. This technique could be compared to the selective scanning algorithm in [18], as in our case most of the time we would also only scan a few channels, the difference being that we have ordered scanning and decide about continuing to scan after every probe response, while they do an increasing order scan of all pre-selected channels. Also note that the selective scanning algorithm of [18] would give good results when only 3 to 4 channels are used in USA, but can give worst results when WLANs have concentrated APs[1] so uses more than 6 channels in USA and can be more worst in most of Europe as total 13 channels can be used in most of Europe while only 11 in USA.

The above ordered selective scanning technique can be used together with neighbor caching algorithm of [18] to completely avoid the search phase, but

[1] Coverage focused WLANs(802.11b/g) have concentrated APs such as at the KTH/Kista campus, while performance focused WLANs have a lower concentration of APs

this works only if the the STA has already been associated to the new-AP before. Again if the WLANs have concentrated APs, then there may be lot of cache misses as most of the time one AP would have more than 2 good neighbors. Also they have used a cache size of 10 entries which would again be a problem in large WLANs where most often the WLAN consists of tens of APs. To overcome this we suggest increasing the size of the cache to 100 entries with a width of 7 to store neighbors. Furthermore, if a STA visits different WLANs then that STA must again build cache for these WLANs even if this STA has been to that WLAN before, but this can be solved if the cache has greater capacity and stores different profiles according to WLAN and selects the cache profile upon first attaching with that WLAN. The neighbor caching algorithm in [18] assumes that APs found in the scan are neighbors of old-AP(current AP), but this algorithm can be improved such that some APs found in the scan could also be neighbors of the new-AP, besides the old-AP.

As the cache contains the possible neighbors of the previous APs to which a STA has been attached with, thus the cache contents can be used in a heuristic algorithm which decides the order of scanning, which we would use if either we have a cache miss or do not have a cache entry for the current AP.

Concerning the flush phase, in our experiments none of the packet got through the flush phase successfully, so we suggest to using it only when two conditions are met, first if the first null function frame(s) (PS sleep trick) before scanning are successful and if the full scanning delay is little bit long, circa more than 50ms, in that case we will probably recover 3 buffered packets.

Until now we have only considered changes in the STA side reducing or avoiding the scanning delay. However, making some changes on AP side would be quite easy and simple to implement, but would requires changes to both AP and STA. This solution would be quite similar the proposed IEEE 802.11k and IEEE 802.11r (see section 1.3.2). The key concept behind that is that AP and STA share a lot of information about the whole WLAN infrastructure and transmission statistics, which may identify neighboring APs and hence help in optimizing handoffs. It also enables the STA to do pre-authentication with neighboring APs.

For IEEE 802.11i enabled WLANs the most efficient solution would be IEEE 802.11k and IEEE 802.11r, but in the meantime we can to take advantage of pre-authentication, for which the neighboring caching technique on STA side (as explained above) needs to be modified a little so in addition to the channel number and MAC addresses of neighboring APs it also stores the capabilities information (RSN IE) or all the beacon information from theses APs. This is needed because pre-authentication needs to know RSN IE of the AP to which it would do pre-authentication.

5.3 Analysis & Improvements – Execution Phase

Apart from the attempted in correct authentication, which is also mentioned by [16], the execution phase went quite smoothly and in accord with expectations. So simply correcting this in correct authentication would save 2 to 3ms from the total handoff latency. This 802.11 authentication is meant to be used as an option when WEP (see section 1.4) is used, but WEP has become obsolete and has been replaced by 802.11i which does not make use of this 802.11 authentication so one suggestion for improving the execution phase would be to totally eliminate this 802.11 authentication or make it optional (in order to be backward compatible), this would save 2 to 3ms. Thus if the AP in its beacon frames announces that 802.11 authentication is optional, then the STA can proceed directly to send a re-association frame instead of an authentication frame.

5.4 Analysis – Detection Phase

Our experiments shows that approximately 70 percent of packets are lost during the detection phase or due to the detection phase. Even though [18] does not talk about the detection phase – it is obvious that once the search phase latency is reduced to a minimum or even avoided all, there is still a considerable amount of packet loss and it is due only to the detection phase. Since most of the packets lost in this phase are consecutive it is quite complex and difficult to recover or conceal this burst loss in higher layers. Obviously users of realtime applications would notice a small spike in their conversation during a handoff, because of the lost packets due to this phase. This is true even if all the improvements suggested above in search and execution phase are in place, therefore it is also very important to reduce the packet loss rate due to this phase.

5.5 Improvements – Detection Phase

As explained before (in section 2.1.4) this phase is a continuous phase and it is very difficult to determine the starting point of this phase. The most common decision for handover initiation is when the STA detects the loss of radio connectivity based on Received Signal Strength Indicator (RSSI) or failed frame transmissions or both. The main difficulty is to determine the reason for the failure, is it due to a collision, radio signal fading, or the station being out of range. Velyos et el. [15] suggest that as soon as the STA experiences frame loss and a collision can be excluded as the reason for frame loss, then the handover phase should start, in this way we would have very little frame loss, only about 3 to 10 frames, which would be easy to recover or conceal in upper layer applications. As detection is crucial for the optimizing the packet loss during the handoff we suggest a much more sophisticated algorithm to detect the start of the handoff process. This algorithm which does not requires any changes of the AP and considers current RSSI, number of failed data frames, number of re-transmitted data frames, number of failed RTS frames, number of re-transmitted RTS frames, number of APs found in last search, or number of neighbors current AP have in STAs neighbor cache. If some changes to the AP side are being made to improve the search and execution phase, then improving this phase on AP side would include exchanging many measurement reports one of which tells the number of STAs attached to the current AP, the concentration of APs in whole WLAN network ,and the number of failed frames (all types) from AP to STA.

5.6 Improvements – Using Cross Layer Communication.

In recent years, the research focus has been to adapting existing algorithms and protocols for compression and transmission of realtime services to the rapidly varying and often-scarce resources of WLANs. However, these solutions often do not provide adequate support for realtime applications in crowded WLANs, when the interference is high or when the STAs are highly mobile. This is because the information which the physical and MAC layers of WLAN have is not shared with the upper layers. This non sharing of information across layers leads to a simple, independent implementation at higher layers, but results in suboptimal performance for realtime application which could perform better by adopting their coding, error correction, compression, and transmission techniques to the current wireless interface conditions. This require applications to work cooperatively throughout the protocol stack.

Though many suggestion provided above would work when physical and MAC layers communicates effectively, but realtime applications could perform even better if the same information which MAC and physical layer share with each other is also shared with upper layers in some standard way. For example handoff could be delayed or done earlier depending on information from higher layers – if any realtime application is in use and if so what is the packet rate, which packets are actually part of realtime traffic, and other similar parameters. Other examples could be allow physical and MAC layer to signal to higher layers that a handoff is going to start. Thus allowing the application to change to a more robust, more handoff friendly codec before the handoff starts. This cross layer communication becomes much more important when mobility is also required at layer-3, by using Mobile IP as described in [12], because the physical and MAC layer could send a signal to layer-3 that a handoff is being made so that it can compete the necessary tasks for layer-3 mobility. However just to know that if the next Layer 2 handoff does require Layer 3 mobility, is much bigger problem so still there is big room for more investigations in order to propose solutions for this problem.

The key concept behind cross layer communication is shown in figure 5.1 which, represents one method of cross layer communication where different layers communicate directly with each other to share the required information.

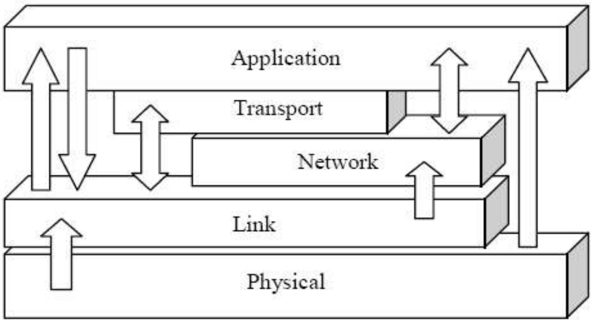


Figure 5.1 - Direct communication method for cross layer communication

6 Conclusions

6.1 Conclusions

In this thesis project we have analyzed each of the WLAN standards and give some insights into the on going standardization work in IEEE 802.11. We throughly examined the handoff process in WLANs and presented handoff latency in its different phases. We also examined the effects of IEEE 802.11i on the overall handoff process. We also presented details on packet loss during the different phases of handoff and presented some techniques for improvements in each phase of handoff. We found that real time applications are most likely to benefit from cross layer communication in WLANs

We did some additional experiments and found some issues which have not previously been described. Fro example how STAs cheat by not scanning all channels, and we presented methods to detect this kind of behavior (even with limited resources).

With respect to IEEE 802.11i we examined out how pre-authentication works, and presented the requirements which must be fulfilled in order to use IEEE 802.11i pre-authentication in IEEE 802.11i enabled WLANs.

Analysis of our experiments showed that there are quite a few methods that can be used to reduce the handoff latency, and we suggested a modified version of selective scanning based on ordered scanning and individual processing of probe responses. We also presented some calculations which show that how these suggestions would reduce handoff latency. We identified detection phase as the next priority. Since so much packet loss occurring in this phase.

We identified cross layer design as the future direction for any wireless network including WLANs as it enables applications to better adapt to the physical environment.

Even though we chose the HostAP driver for our experiments, we learned about another driver[26] which implements the whole Hardware Abstraction Layer (HAL) in software hence does not have the limitations which HostAP has. We suggest using that driver in future research on handoff and related issues in WLANs

Although initially I intended to do all the experiments in actual WLANs also (some factors such as load on the APs and interference are not present in testbed and these might influence the handoff), but due to limited time I was not able to do this.

6.2 Future work

Due to limitations in the HostAP software and my time, we did not implement any of our suggested improvements, so the most obvious future work would be to first implement the suggestions presented in sections 5.2, 5.3, and 5.5 – specifically those requiring changes only on the STA side as they are likely to be the easy to implement.

There is considerable future work in “cross layer optimization” focusing on issues and solutions using a cross layer architecture.

7 References

[1] IEEE Std 802.11, 1999 Edition, **"IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications."**
http://standards.ieee.org/getieee802/802.11.html

[2] IEEE 802.11b -1999 **"Supplement to 802.11-1999,Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band"**
http://standards.ieee.org/getieee802/802.11.html

[3] IEEE 802.11a-1999, **"IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless – LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 1: High-speed Physical Layer in the 5 GHz band."**
http://standards.ieee.org/getieee802/802.11.html

[4] IEEE 802.11g-2003, **"IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band."**
http://standards.ieee.org/getieee802/802.11.html

[5] IEEE 802.11d-2001, **"Amendment to IEEE 802.11-1999, (ISO/IEC 8802-11) Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Operation in Additional Regulatory Domains."**
http://standards.ieee.org/getieee802/802.11.html

[6] IEEE 802.11h-2003, **"IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Spectrum and Transmit Power Management Extensions in the 5GHz band in Europe**
http://standards.ieee.org/getieee802/802.11.html

[7] IEEE 802.11j-2004, **"IEEE Standard for Information Technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications."**
http://standards.ieee.org/getieee802/802.11.html

[8] IEEE 802.11i-2004, **"IEEE Standard for Information technology – Telecommunications and information exchange between system – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications--Amendment 6: Medium Access Control (MAC) Security**

Enhancements.”
http://standards.ieee.org/getieee802/802.11.html

[9] ZDNet UK News, **“Intel hangs mesh hopes on 802.11s”** – March 3, 2005 http://news.zdnet.co.uk/communications/wireless/0,39020348,39189953,00.htm

[10] IEEE 802.21, **“Media independent handover and interoperability between heterogeneous networks.”**
http://www.ieee802.org/21

[11] IEEE 802.11 WG current activities & timelines, **“Official IEEE 802.11 working group project timelines”** – May 20, 2005
http://grouper.ieee.org/groups/802/11/802.11_Timelines.htm

[12] J-O. Vatn, **“IP telephony: Mobility and security”** Doctoral thesis, Institute of Microelectronics and Information Technology, KTH, Stockholm, SWEDEN. Pages: 79 -118, June 2005
http://www.diva-portal.org/kth/theses/abstract.xsql?dbid=260

[13] Monsters and Critics, Tech News, **“VoIP Expansion to Challenge Networks”** – April 30, 2005
http://tech.monstersandcritics.com/news/article_7152.php/VoIP_Expansion_to_Challenge_Networks

[14] A. Mishra, W. Arbaugh, and M. Shin **“An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process”** ACM SIGCOMM Computer Communication Review., Vol: 32, Issue: 2, Pages: 93 – 102, April 2003.
http://www.cs.umd.edu/~waa/pubs/handoff-lat-acm.pdf

[15] H. Velayos, and G. Karlsson, **“Techniques to Reduce IEEE 802.11b MAC Layer Handover Time”** IEEE International Conference on Communications., Vol. 7, pages. 3844 – 3848, June 20 - 24, 2004.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?isnumber=29132&arnumber=1313272&count=119&index=13'

[16] J-O. Vatn, **”An experimental study of IEEE 802.11b handover performance and its effect on voice traffic”**, KTH, Royal Institute of Technology, Kista, SWEDEN, July 2003.
http://www.it.kth.se/~vatn/research/handover-perf.pdf

[17] F. Mlinarsky, and G. Celine, **“Testing Braces Itself For Voice Over Wi-Fi -
As long as wireless test can depict its roaming time and behavior, VoWi-Fi Seems Fated To Inherit Wi-Fi's Popularity”** - Wireless Design Articles, Product Summaries, and Application Notes for Wireless Design Engineer, July/August 2004.
http://www.wsdmag.com/Articles/ArticleID/8627/8627.html

[18] S. Shin, A. S. Rawat, and H. Schulzrinne. **“Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs.”** ACM MobiWac'04 , Philadelphia, Pennsylvania, USA – Oct 2004

[19] S. Park and Y. Choi. “**Fast inter-AP handoff using predictive authentication scheme in a public Wireless LAN.”** Networks - 2002 (Joint ICN 2002 and ICWLHN 2002), August 2002.
http://mmlab.snu.ac.kr/~webmaster/publications/docs/Networks2002_shpack.pdf

[20] S. Park and Y. Choi. **“Pre-authenticated fast handoff in a public Wireless LAN based on IEEE 802.1x model.”** IFIP TC6 Personal Wireless Communications – Oct 2002.

http://mmlab.snu.ac.kr/~webmaster/publications/docs/pwc2002_shpack.pdf

[21] H. Kim, S. Park, C. Park, J. Kim, and S. Ko. **“Selective Channel Scanning for Fast Handoff in Wireless LAN using Neighbor Graph.”** ITC-CSCC 2004 - July 2004.
http://dali.korea.ac.kr/publication/int_pro/paper/IntPro085.pdf

[22] Open Source WLAN driver. **“Host AP driver for Intersil Prism2/2.5/3, hostapd, and WPA Supplicant”** - Last visited: 15th May 2005
http://hostap.epitest.fi/

[23] Open Source RADIUS server. **“The FreeRADIUS Server Project”** - Last visited: 15th May 2005
http://www.freeradius.org/

[24] Prism 2/2.5/3 Chipset manual **“PRISM Driver Programmers Manual”**
Last visited: 31st January 2005
http://home.eunet.cz/jt/wifi/RM0251.pdf

[25] MGEN **“The Multi-Generator Toolset”** - Last visited: 17th May 2005
http://mgen.pf.itd.nrl.navy.mil/

[26] Linux driver for 802.11a/b/g WLAN cards using Atheros chip sets **“Multiband Atheros Driver for WiFi (MADWIFI)”** - Last visited: 17th May 2005
http://sourceforge.net/projects/madwifi/

8 Appendices

Appendix A: Abbreviations and Acronyms

AAA	Accounting, Authorisation, Authentication
AP	WLAN Access Point
BSSID	Basic Service Set IDentity
DCS	Dynamic Channel Selection
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
EAPoL	Extensible Authentication Protocol over Local Area Network
ESSID	Extended Service Set IDentity
FHSS	Frequency Hopping Spread Spectrum
IAPP	Inter Access Point Protocol
IEEE	Institute of Electrical and Electronic Engineers
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
OTA	Over The Air
PDA	Personal Digital Assistant
PMK	Pairwise Master Key
PMKSA	Pairwise Master Key Security Association (defined in IEEE 802.11i [8])
PS	Power Save
QoS	Quality of Service
RP	Recommended Practice
RSNIE	Robust Security Network Information Element (defined in IEEE 802.11i [8])
STA	WLAN Client Station
TDMA	Time Division Multiple Access
TPC	Transmission Power Control
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network
VoIP	Voice over Internet Protocol

Appendix B: Configurations and Settings

Installation of HostAP suite on Fedora Core3.

HostAP suite comes with four components

```
hostap-driver:
      Linux driver for Intersil Prism2/2.5/3
      Host AP mode support, Limited IEEE 802.11 management (AP)

hostapd:
      User space daemon for extended IEEE 802.11 management
      IEEE 802.1X Authenticator
      WPA/WPA2 Authenticator
      RADIUS Authentication client
      RADIUS Accounting client

hostap-utils:
      Utility programs for hostap-driver

wpa_supplicant:
      Additional program needed for WPA and WPA2 client
      operations. The Supplicant is used in WPA/WPA2 key
      handshakes to authenticate with the AP and to generate
      dynamic encryption keys (TKIP or CCMP).
```

`hostap-driver` and `hostap-utils` are installed on every node while `hostapd` is installed on Access Points (AP1 and AP6) and `wpa_supplicant` is installed on STA.

Bridge configuration. (for AP1)

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 wlan0
ifconfig br0 10.10.1.11 up
iwconfig wlan0 channel 1
ifconfig wlan0 up
ifconfig eth0 10.10.1.1 up
route add default eth0
```

hostapd.conf file for Test in section 4.1.1 (for AP1)

```
interface=wlan0
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=0
debug=4
dump_file=/tmp/hostapd.dump
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
ssid=test
macaddr_acl=0
auth_algs=1
ieee8021x=0
iapp_interface=eth0
```

hostapd.conf file for Test in section 4.1.2 (for AP1)

```
interface=wlan0
logger_syslog=-1
```

```
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=0
debug=4
dump_file=/tmp/hostapd.dump
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
ssid=test
macaddr_acl=0
auth_algs=1
ieee8021x=1
eap_message=hello
eapol_key_index_workaround=0
eap_reauth_period=400
iapp_interface=eth0
own_ip_addr=10.10.1.1
nas_identifier=ap1
auth_server_addr=10.10.1.3
auth_server_port=1812
auth_server_shared_secret=secret-ap1
acct_server_addr=10.10.1.3
acct_server_port=1813
acct_server_shared_secret=secret-ap1
wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=TKIP CCMP
rsn_preauth=1
rsn_preauth_interfaces=eth0
```

AP6 Configuratrions:

Bridge and interfaces configuration. (for AP6)

```
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 wlan0
ifconfig br0 10.10.1.66 up
iwconfig wlan0 channel 1
ifconfig wlan0 up
ifconfig eth0 10.10.1.6 up
route add default eth0
```

hostapd.conf file for Test in section 4.1.1 (for AP6)

```
interface=wlan0
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
logger_stdout_level=0
debug=4
dump_file=/tmp/hostapd.dump
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
ssid=test
macaddr_acl=0
auth_algs=1
ieee8021x=0
iapp_interface=eth0
```

hostapd.conf file for Test in section 4.1.2 (for AP6)

```
interface=wlan0
logger_syslog=-1
logger_syslog_level=2
logger_stdout=-1
```

```
logger_stdout_level=0
debug=4
dump_file=/tmp/hostapd.dump
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
ssid=test
macaddr_acl=0
auth_algs=1
ieee8021x=1
eap_message=hello
eapol_key_index_workaround=0
eap_reauth_period=400
iapp_interface=eth0
own_ip_addr=10.10.1.6
nas_identifier=ap6
auth_server_addr=10.10.1.3
auth_server_port=1812
auth_server_shared_secret=secret-ap6
acct_server_addr=10.10.1.3
acct_server_port=1813
acct_server_shared_secret=secret-ap6
wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=TKIP CCMP
rsn_preauth=1
rsn_preauth_interfaces=eth0
```

STA Configuratrions:

Interfaces configuration.

```
iwconfig wlan0 mode managed
ifconfig wlan0 10.10.1.20 up
route add default wlan0
```

wpa_supplicant.conf file for Test in section 4.1.2

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=2
ap_scan=1
fast_reauth=1
network={
      ssid="test"
      proto=RSN
      key_mgmt=WPA-EAP
      auth_alg=OPEN
      pairwise=CCMP
      group=CCMP
      eap=TLS
      identity="example-user"
      ca_cert="/usr/ffdownloads/freeradius-certs/cacert.pem"
      private_key="/usr/ffdownloads/freeradius-certs/cert-
clt.p12"
      private_key_passwd="whatever"
}
```

Traffic generation using MGEN for Test in section 4.2.1

```
mgen event "ON 1 UDP DST 10.10.1.3/7000 PERIODIC [50.0 172]"
```

Traffic receiving using MGEN for Test in section 4.2.2

```
mgen event "listen udp 7000" output ReceivedSTA1.txt
```

Traffic generation and receiving using MGEN for Test in section 4.3

```
mgen event "ON 1 UDP DST 10.10.1.3/7000 PERIODIC [50.0 172]"
mgen event "listen udp 7001" output ReceivedSTA2.txt
```

Host-R Configuratrions:

Apart from `hostap-driver, freeradius` server is also installed on this host

Interfaces configuration.

```
ifconfig eth0 10.10.1.3 up
iwconfig wlan0 channel 1
iwconfig wlan0 mode monitor
iwconfig wlan1 channel 6
iwconfig wlan1 mode monitor
ifconfig wlan0 up
ifconfig wlan1 up
```

Traffic receiving using MGEN for Test in section 4.2.1

```
mgen event "listen udp 7000" output ReceivedHostR1.txt
```

Traffic generation using MGEN for Test in section 4.2.2

```
mgen event "ON 1 UDP DST 10.10.1.3/7000 PERIODIC [50.0 172]"
```

Traffic generation and receiving using MGEN for Test in section 4.3

```
mgen event "ON 1 UDP DST 10.10.1.20/7001 PERIODIC [50.0 172]"
mgen event "listen udp 7000" output ReceivedHostR2.txt
```

Additions in clients.conf file (/etc/raddb/clients.conf) for Test in section 4.1.2

```
client 10.10.1.1
{
      secret  = secret-ap1
      shortname       = AP1
}

client 10.10.1.6
{
      secret  = secret-ap6
      shortname       = AP6
}
```

Additions in eap.conf file (/etc/raddb/eap.conf) for Test in section 4.1.2

```
tls
{
      private_key_password = whatever
      private_key_file = ${raddbdir}/certs/cert-srv.pem
      certificate_file = ${raddbdir}/certs/cert-srv.pem
      CA_file = ${raddbdir}/certs/cacert.pem
      dh_file = ${raddbdir}/certs/dh
      random_file = ${raddbdir}/certs/random
      fragment_size = 1024
      include_length = yes
}
```

Commands used for experiments in section 4.4 and for setting the environment.

To turn on radio header capture.

```
iwpriv wlan0 monitor_type 2
```

To scan perticular set of channels
(e.g, to scan channel 6 at 1Mbps)

```
hostap_rid wlan0 set fce5 02 00 00 0a 00 00
```

for more information see
(section 4.4.2.2.29 of [24])

To set output power

`iwpriv wlan0 alc 0` **(disabling ALC = Automatic level Control)**

`iwpriv wlan0 writemif 62` *`$x`*

($x = 0 to 255 where 127 is the lowest and 128 is the highest)
for more information see
(http://www.hasw.net/wlan/NL-2511-TxPower.png)

Sections from [24] which may be useful in future work:

4.4.2.1

4.4.2.1.7 (last para most important)

4.4.2.1.11

4.4.2.1.17

4.4.2.1.25

4.4.2.1.30

4.4.2.2.1

4.4.2.2.2

4.4.2.2.9

4.4.2.2.17

4.4.2.2.20

4.4.2.2.25

4.4.2.2.26

4.4.2.2.28

4.4.2.2.29

4.4.3.1

4.4.3.1.10

4.4.3.1.11

4.4.3.2 (Table 4.17)

4.4.3.2.1 (I think this one influences the handoff decision)

4.4.3.2.4 (I think this one influences the handoff decision)

4.4.3.2.7 (I think this one influences the handoff decision)

4.4.3.2.16

4.4.3.2.19

4.4.3.2.20

4.4.4.1 (All Tables)

4.4.4.2 (All Tables)

4.4.5.2.2

4.4.5.2.3

4.4.5.2.4

4.4.5.3

IMIT/LCN 2005-19

www.kth.se