VoWiFi Roaming

SYED MUHAMMAD ALI

KTH Information and
Communication Technology

Master of Science Thesis
Stockholm, Sweden 2006

COS/CCS 2006-2

VoWiFi Roaming

Syed Muhammad Ali
masyed@kth.se

23 Jan 2006

School of Information and Communication Technology (ICT)
Royal Institute of Technology (KTH)

Thesis Advisor and Examiner: Gerald Q. Maguire Jr.
Industrial supervisor: Rasmus Axén

Sponsor: Ericsson AB , Sweden

Abstract

Freedom is human's natural instinct, which was *limited* by Ethernet and Fixed Telephony *Era*. With the emergence of new technologies like wireless fidelity (WiFi) and voice over IP (VoIP) humans once again have freedom of movement; which at the very same time provides *enough reasons* to change the market dynamics of communication industry. The buzz of Voice over WiFi (VoWiFi) in recent years indicates that VoWiFi is shaping up as the next big challenge to traditional telephony, not only due to cost, but also due to range of services and amount of freedom it can offer. However, at the very same time these technologies have evolved to threaten the *well-established* telephony markets. Enterprise solutions for VoWiFi require enhanced security mechanism and seamless handovers. To address security related issues Wi-Fi Alliance in conjunction with IEEE introduced an enhanced and interoperable security scheme called WiFi Protected Access (WPA).

Real time services are sensitive to latency, hence requiring bounded delay time throughout an ongoing session. Handovers in WiFi networks can take fairly long time which real time services cannot tolerate. The problem is further elevated when WiFi networks are secured by using WPA Enterprise.

In this thesis we will examine the complete handoff process in WiFi networks. The impact of handovers on VoIP traffic will also be observed. Following the detailed analysis some suggestions will be presented concerning how to reduce this handoff latency.

Sammanfattning

Friheten som ligger i människans natur begränsades av Ethernet och den fasta telefonin. Med uppkomsten av nya teknologier så som Wireless Fidelity (WiFi) och Voice over IP (VoIP) återfår människan den en gång förlorade friheten. Samtidigt kommer telekommunikationsindustrin att kunna ändras till sin struktur genom WiFi och VoIP.

Integreringen av Voice over IP och WiFi , även mer känd som Voice over WiFi, (VoWiFi) har under senare år indikerat att det är en potentiell utmanare till traditionell telefoni inte bara ur ett kostnadsperspektiv utan också för att denna teknologi medför ökade möjligheter när det gäller nya tjänster. Dock återstår en del arbete för VoWiFi för att kunna rubba den fasta telefonin. Företagslösningar av denna teknologi kräver att säkerhetsaspekterna ses över dessutom måste seamless handover fungera på ett tillfredställande sätt. För att se över säkerhetsaspekterna har Wi-Fi Alliance i samarbete med IEEE introducerat säkehetsmekanismen WiFi Protected Access (WPA).

Realtidstjänster är känsliga mot fördröjningar. Handover i ett WiFi nätverk kan ta relativt lång tid vilket är oacceptabelt för realtidstjänster. Problemet blir än mer påtagligt när WiFi-nätet är säkrat med hjälp av WPA.

I denna exjobbsrapport kommer handoff processen för WiFi nätverk att behandlas. Effekten av handover för VoIP trafik kommer också att beskrivas. Resultat och analyser kommer att föreslås för hur man kan reducera handoff-fördröjningar.

Acknowledgments

First of all I am grateful to my God for His help, then I would like to thank my supervisor at KTH, Professor Gerald Maguire who really guided me all the way. He was always there, whenever I needed him. Professor Maguire is much more than a thesis supervisor to me. I *believe* that, it is simply not possible for me to acknowledge his support, in words here.

I am very grateful to my supervisor at *Ericsson,* Rasmus Axén who was very kind and encouraging throughout the thesis work. I was very lucky to have Bo Kvarnström as my co-supervisor at *Ericsson*, who was very generous in sharing his vast experiences. Johan Danestig was very cooperative in solving various *Azimuth Systems* related issues despite his busy schedule. Peter Nilson was very valuable resource in software design team, who was always ready to help. Sam Fong, Björn Manholm and Thorsten Rhau were also very helpful. I am grateful to Helene Rödjevi and Zaid Karlsson for solving all administrative issues. Thanks to Niclas Nors and the whole *Ericsson CPEPS I&V department* for their help, without which it wasn't possible. Thanks to my colleague Jia Liu for her support during the thesis work.

I would like to thank Anders Lindström and Tom Idemark, *Manager CPEPS department* and *Ericsson GSM and Radio Access networks Region,* Linköping, Sweden for providing me opportunity to explore this endless voyage of knowledge.

I am grateful to J.O.Vatn and Ajeet Nankani for sharing their experiences on the subject. I am very grateful to Johan Montelius (*Programme coordinator, Internetworking, KTH*) for his valuable feedback and support throughout the course.

In the end I would like to thank my family for believing in me, and supporting me all the way.

To

My Mentor

‘’Syed Ameer shah Qadri Gillani (Rahmatullah 'alaih) ‘’

Who was always with me.

TABLE OF CONTENTS

1 Introduction

1.1 Problem statement:

VoIP is changing the paradigm of the communication industry -- forcing it to change from circuit switched to end-to-end packet switching. An advantage of using packet-based communications is that several multiple access technologies can be utilized. Fast handover between different access technologies is the next 'Big Thing' in the communication world. Although technologies for carrying voice over IP networks have been advancing for quite some time, limitations of carrying voice over wireless packet networks means it will be some time before they are serious threat to the traditional telecom market. Packet loss due to long handover delays, security concerns and lack of support for guaranteed throughput i.e. Quality of Service (QoS) at medium access layer (MAC) layer, are few of these limitations. To address the QoS issue IEEE introduced a new standard called 802.11e [17], while the IEEE 802.11i addresses the security related issues in WLAN.

This thesis focuses on how handover within an 802.11 network affects voice applications in terms of performance. Different types of 802.11 network scenarios will be considered when measuring handover delays. The criteria for starting and completion of a handover will also be defined during the thesis.

The expected handover delays will be calculated and compared with actual measurements. Any deviations between the results and expectations should be investigated and described. An Azimuth system [26] was used to perform some measurements, but it was complemented with other measurements tools .

1.2 Voice over IP (VoIP)

Voice over IP is defined as voice delivered using the Internet Protocol [2].VoIP operates by digitizing voice, encapsulating it in IP packets, sending those packets over an IP network, and eventually converting the packets back to audio for the *callee* to hear at the other end of the line. Subsequently, the process is repeated in reverse, so one can hear the other person's voice. In an ordinary fixed line phone call, voice is turned into a 64kbps digital bit stream. This digital voice channel is multiplexed and transported and eventually demultiplexed at the other end. All along the path these bits have been circuit switched along a *pre-selected* path. For the duration of the call, the caller is assigned a fixed bandwidth 64kbps channel along the entire physical path **and** the reverse path.

Today 64kbps channel is more then enough for voice, when limited to 3.3 kHz bandwidth used telephony. Research has shown that very good-quality encoding is possible at 8 to 12 kbps. This is commonly called *compressed voice*, as it uses far less than the conventional 64 kbps. At 8 kbps, one could pack eight phone calls in *each* 64kbps conventional channel's bandwidth [3]. At 8 kbps, one can also send that digitized signal over the Internet with very little impact on the network. However, savings in terms if bandwidth are one aspect, VoIP also opens up a broad range of other services benefiting from packet-based networks like Location Based Services, Voice portals for Interactive Voice Responses multimedia conference calls, etc.

1.2.1 VoIP Transmission

One problem with sending voice over the Internet is that sequential packets sometimes take different paths to reach the same destination and they may also face different delays. This doesn't cause problems while sending files, but is a problem if the data packets consist of digitized voice. To operate properly, voice packets should be sent with limited loss and minimal delay. The codec may label each packet identifying it as voice (implying a higher priority). Transport protocol is used to identify the order of the packets and when they were sent, thus the receiver can resequence the received packets, if necessary, and buffer them in such a way that they can be decoded and output to the digital-to-analog converter with the correct timing. Otherwise, the far end would receive very choppy, distorted, voice with annoying gaps and delays. One method of improving this situation is to provide a path over the Internet that explicitly supports a specific quality of service (QoS) [6]. The IEEE 802 Ethernet protocols already have a provision for denoting and maintaining QoS, by utilizing the recently approved standard 802.11e. [17]

1.2.2 Session Initiation Protocol (SIP)

1.2.3 Introduction to SIP

There are many Internet applications that require the creation and management of a session, where a session is involves an exchange of data between a group of participants. The implementation of these applications is complicated as: users may move between endpoints, they may be addressable by multiple names, and they may communicate in different media (sometimes simultaneously). Numerous protocols have been designed to carry various forms of real-time multimedia session data such as voice, video, or text messages. The Session Initiation Protocol (SIP)[7] works in concert with these protocols by enabling internet endpoints (called user agents) to discover one another and to agree on characteristics of a session. To locate prospective session participants, and for other functions, SIP utilizes an infrastructure of network hosts (called proxy servers) to which user agents send registrations, invitations to sessions, and other requests. SIP is an agile, general-purpose tool for creating, modifying, and terminating sessions that works independently of underlying transport protocols and without regard to type of session that is being managed.

1.2.4 Overview of SIP Functionality

SIP is an application-layer control protocol that can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP can invite participants to existing sessions, such as multicast conferences. Media can be added to (and removed from) an existing session. SIP transparently supports name mapping and redirection services, which supports personal mobility- thus users can maintain a single externally visible identifier regardless of their network location.

SIP supports five facets of establishing and terminating multimedia communications:
User location is used in determining the end system to be used for communication.

User availability is used to determine the willingness of the called party to engage in communications. *User capabilities* are used in determining the media and media parameters to be used. SIP uses *Session Set up* for the session initiation, i.e. the mutual establishment of session parameters at both called and calling party while *Session management* is used for invoking services transfer, terminate the sessions, and to modify the session parameters.

SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method, or function, on the server and generates at least one response. A SIP URI (Uniform Resource Indicator) identifies a communications resource. In this example, the transaction begins with Alice's soft phone sending an INVITE request addressed to Bob's SIP URI. INVITE is an example of a SIP method that specifies the action that the requestor (Alice) wants the server (Bob) to take. The INVITE request contains a number of header fields. Header fields are named attributes that provide additional information about a message. An INVITE includes a unique identifier for the call, the destination address, Alice's address, and information about the type of session that Alice wishes to establish with Bob. The time line of an INVITE is shown in figure 1.1

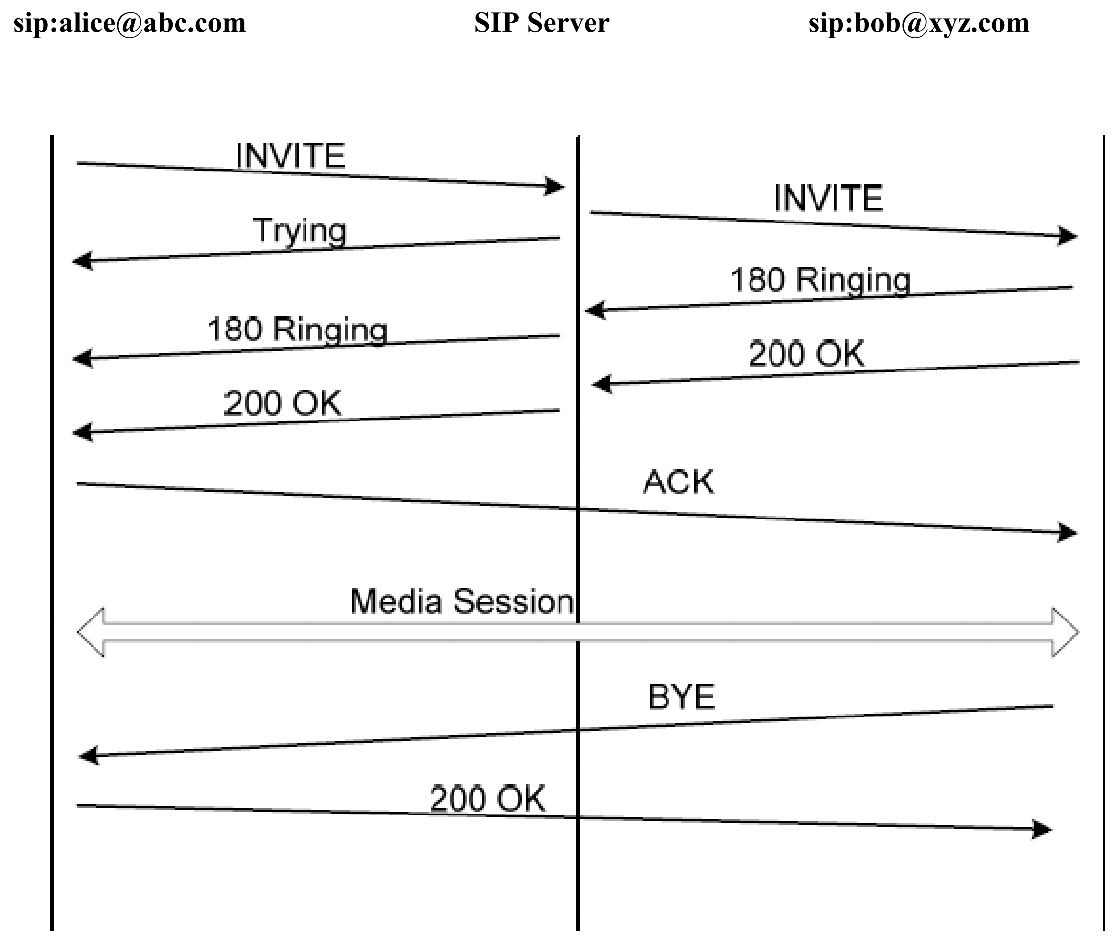


Figure 1.1: SIP session set up example with SIP trapezoid

After an INVITE, a number of messages are exchanged to setup the media session between Alice and Bob. Finally a BYE message terminates the call.

1.2.5 SIP Components

SIP basically has two components:

1. SIP User Agents
2. SIP Network Servers

The User agent exists in the end system and consists of two parts:
(a) The client element called User Agent Client (UAC) is used to initiate a call;
(b) The server element, called the User Agent Server (UAS) is used to answer requests.

The SIP servers' functions include resolving the URI and determining the user's locations. The caller does necessarily know the IP address or even the hostname of the called party. The following are examples of SIP servers:

Registrar server

The registrar server receives Register requests from the UAC's. The Register request associates the user's SIP address, called a SIP Uniform Resource Identifier (URI), with the current address, where the user can be located. The Location Service (LS) stores this association. It is important to note that the location of the user and the location of their UA need not be the same

Proxy Server

Callers can send their SIP requests via a Proxy Server, which forwards the requests to the next hop proxy server or to a proxy server close to the called user. The proxy server can modify or add information to SIP requests. A Domain Name System (DNS) Server can be used to find the location of the Proxy server.

Redirect Server

The Redirect server receives requests from clients, but unlike Proxy Servers, it does not forward the request to another server or the user. Rather, it sends back a response to the requester with the information about the destination.

1.2.6 SIP Addresses

SIP users are identified by SIP addresses, called a SIP URI. The SIP URI looks like an email address, i.e., username@somedomain, where the first part is the username or a phone number and the second part is the domain name or the network address [4]. An example of a SIP address would be "sip:ali@sip.ericsson.com" where "ali" is the username and "sip.ericsson.com" is the domain name. SIPS indicates secure SIP URI introduced in RFC 3261 [2] and it *requires* that a secure mechanism be used between the user agent and the proxy the user is contacting.

1.2.7 Session Description Protocol (SDP)

SIP is not meant to provide services and hence uses other protocols to provide services and media related information, e.g., specific CODEC, and other media parameters. For this purpose, SIP uses the Session Description Protocol (SDP) [5], which conveys the information about media streams in multimedia sessions. The media related information such as type of media (video or voice) and type of CODECs, etc. is transmitted in a simple textual format called the SDP body and is added to the body of the SIP INVITE messages when a call is initiated. This informs the called party of the session parameters acceptable to the calling party. Adding the SDP body to a SIP INVITE message avoids generating unnecessary traffic and reduces the call setup delay as the parameters can be communicated at the same time as the call setup. The reply from the called party describes the selected session related capabilities [5] [6].

1.3 Wireless LAN

1.3.1 802.11standards

In 1997, the Institute of Electrical and Electronics Engineers (IEEE) adopted the first WLAN standard. It is often called IEEE 802.11, after the name of the working group formed to oversee its development. Initially 802.11 only supported a maximum bandwidth of 2 Mbps. Later on a series of standards were defined to improve performance of such WLAN's. Several of these standards (presented in alphabetical order) are :

Standard	Description
802.11a	Operates in the 5Ghz band, data rates up to 54 Mbps
802.11b	Operates in the 2.4 Ghz band, data rates up to 11 Mpbs.
802.11e	Enhances 802.11 MAC to improve QoS for real-time services.
802.11f	Inter-Access Point Protocol; increases compatibility between Access Point devices from multiple vendors.
80.2.11g	Operates in the 2.4 Ghz, and data rate up to 54 Mbps, compatible with 802.11b devices.
802.11h	Enhances to provide network management and control extensions for spectrum and transmit power management in the 5 GHz band.
802.11i	Enhances the security and authentication mechanisms
802.11k	Radio Resource Measurements
802.11n	Proposed standard, data rates up to 540 Mbps
802.11r	Inter-AP handoffs (Fast Roaming)

Table 1.1: 802.11 Standards

1.3.2 WLAN Protocol Layers and sub layers

The IEEE 802.11 standard defines physical (PHY) and Medium access Control (MAC) sub-layers for WLAN along with their relation to higher layers specifically IEEE 802.2. These relations are illustrated in figure 1.2.

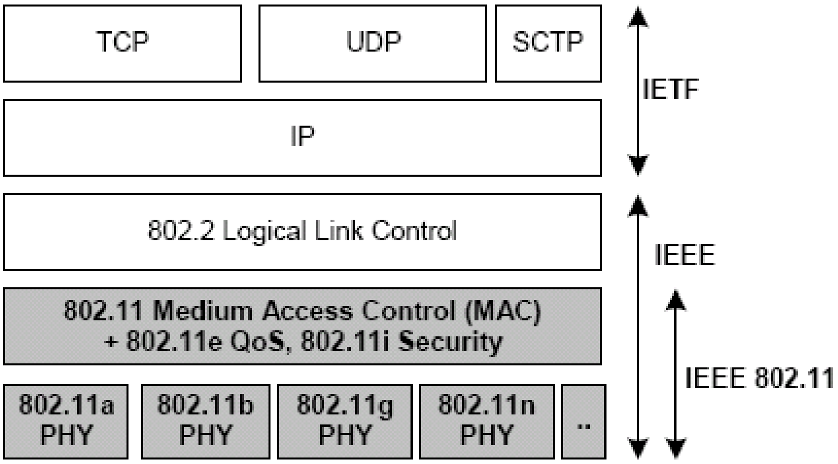

TCP
UDP
SCTP
IETF
IP
802.2 Logical Link Control
IEEE
802.11 Medium Access Control (MAC)
+ 802.11e QoS, 802.11i Security
IEEE 802.11
802.11a PHY
802.11b PHY
802.11g PHY
802.11n PHY
..

Figure 1.2: IEEE 802.11 layers and sub layers [21]

1.3.3 WLAN Architecture

The IEEE 802.11 architectures can be divided into *infrastructure* and *ad hoc* architectures. In *ad hoc* mode, a mobile station works independently and communicates directly with others when in signalling range. In *infrastructure* mode, each mobile station will connect to an Access Point, which acts as a Base Station that connects between mobile stations and another wired or wireless network.

1.3.4 802.11b, 802.11a and 802.11g

The IEEE 802.11 working group defined various WLAN standards. IEEE 802.11b is based on direct sequence spread spectrum (DSSS) technology, as opposed to 802.11a, which is based on orthogonal frequency-division multiplexing (OFDM). The later provides higher data rates; 802.11b can reach 11 Mbps, while 802.11a can reach 54 Mbps. Vendors often quote both of these figures, but they are a bit misleading. The physical layer overhead cuts throughput by at least 40 percent, meaning the actual user rate of 802.11b is at most around 6 Mbps. In practice, it's a lot less. As 802.11a and 802.11b WLANs use unlicensed spectrum; they're prone to interference and the usual transmission errors. These errors may mean that traffic has to be resent, which wastes bandwidth. A 50 percent error rate will reduce the real throughput by about two-thirds, to only 2 Mbps; furthermore the channel is shared by every node on the network. To reduce errors, both types of 802.11 can automatically reduce their data rate. IEEE 802.11b has three lower data rates (5.5, 2, and 1 Mbps), and 802.11a has seven (48, 36, 24, 18, 12, 9, and 6 Mbps), actually the 802.11a has the same physical signalling as for 802.11b. The lower rates are used most of the time[9]. Higher data rates are not the only advantage of the 802.11a. It also uses a higher frequency band, i.e. the 5 GHz, which is both wider and less crowded than the 2.4 GHz band that 802.11b shares with cordless phones (only in US), microwave ovens, and Bluetooth devices. The wider band means that more radio channels can coexist without interference. Each radio channel corresponds to a separate network, or a switched segment of the same network. The precise number of channels varies by country, because regulators have allocated a different amount of spectrum for unlicensed use in different countries. However, there are always more channels in the 5GHz band than the 2.4 GHz band. In the United States, the 2.4GHz band is wide enough for only three, whereas 5 GHz has room for 11. Although 5 GHz has many advantages, it also has some problems. The most significant of these is compatibility: The different frequency means that 802.11a products aren't interoperable with the installed 802.11b base. To get around this IEEE 802.11 *Task Group "G"* approved a wireless data local-area network standard that provides data rates up to 54 Mbps in the 2.4 GHz frequency band. IEEE 802.11g attempts to combine the best features of both 802.11a and 802.11b, thus the 802.11g is based on OFDM operates in the 2.4GHz band and provides the same coverage as 802.11b. Unfortunately, interference means that it will never be as fast as 802.11a [9]

1.3.5 802.11e

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 provides classes of service with managed levels of QoS for data, voice, and video applications. The IEEE 802.11e enhances the IEEE 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 was recently ratified in 2005 and should start appearing in products this early 2006.

1.3.6 WLAN Coordination Function

A basic service set (BSS) is defined as a set of stations controlled by a single coordination function the coordination function is a logical function that determines when a station operating within a basic service set (BSS) is permitted to transmit. The coordination function within a BSS may either be the point coordination function (PCF) or distributed coordination function (DCF).

Point Coordination Function (PCF)
In this class of coordination functions one station in a basic service set (BSS) is operates at any given time and one node controls all of the other stations within the basic service set.

Distributed Coordination Function (DCF)
In this class of coordination function the same coordination function logic is active in every station in the BSS. Each station independently executes this same function. ; Thus there is no single node that controls the "cell".

The basic medium access protocol is a DCF that allows for automatic medium sharing between compatible PHYs through the use of CSMA/CA with a random backoff time following a busy medium condition. In addition, all unicast traffic uses immediate positive acknowledgments (ACK frames), thus the sender will schedule retransmission if an ACK is not received. [1]

1.3.7 Timing

IP packets travel from the WLAN client through the wireless network to the IP backbone, then these IP datagrams are first encapsulated into link frames and later into radio frames, later they are decapsulated back to IP datagrams at the access point *used in routing mode*. The IEEE 802.11 specifies the Distributed Coordination Function (DCF) as the default media access control (MAC) mechanism for WLAN wireless networks. The DCF is composed of two main components:

1. Interframe space (IFS) and
2. Random backoff (contention window)

Interframe space (IFS)
Use of an IFS allows 802.11 to control which node gains access to the radio channel once the absence of a carrier indicates that the channel is free. High priority 802.11 management and control frames use the Short IFS (SIFS) spacing to have the fastest access to the media. Most other data frames wait the Distributed IFS (DIFS) before attempting to gain radio access for transmission.

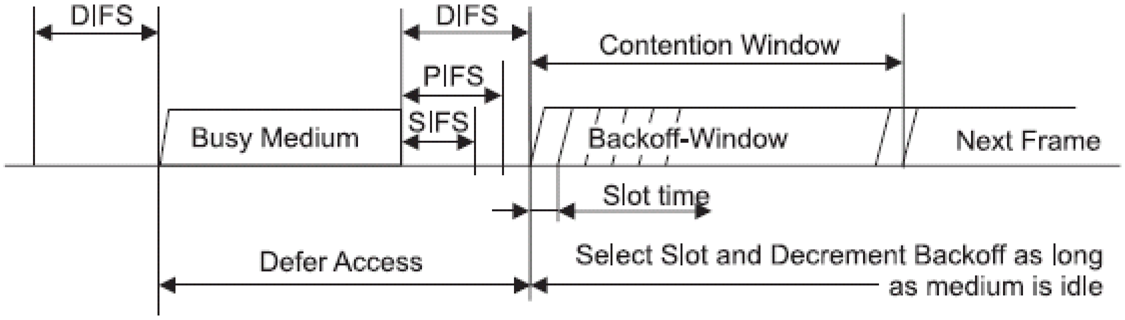


Figure 1.3: Interframe spacing and Contention

Random Backoff (contention window)

DCF uses a random backoff algorithm to avoid collisions in the radio channel (hence the protocol is CSMA/CA). The value of the random backoff timer is controlled by a contention window (CW), which is defined as a value between CW_{min} and CW_{max}. Initially, the backoff timer is a random number between 0 and CWmin. It decrements every 20 µs (the slot time) during which the radio channel remains free. A data frame can be sent only when the available radio channel remains free after the backoff timer reaches zero. However, if the data frame is not sent before the initial random backoff timer expires, the WLAN client or access point will increment the retry counter and restart the process with a new random backoff window, doubled in size. This doubling in size will continue until the final window size equals CW_{max}. The retries continue until the maximum retries or time-to-live (TTL) have been reached. DCF mainly defines the MAC protocol for WLAN wireless networks. Other than 802.11 management and control frames, DCF alone does not provide traffic prioritization directly to other data frames. [10]

2 Background

2.1 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.

Initially the 802.11 WLAN standards included a security protocol called Wired Equivalent Privacy (WEP), which was designed to protect frames data packets well enough to keep out causal eavesdroppers. WEP encrypts each 802.11 frame separately with an RSA RC4 cipher stream generated by a 64-bit RCA key. However, several cryptanalysts have identified weaknesses in the RC4 key scheduling algorithm that makes the network vulnerable to hackers. Software tools such as AirSnort [18] have been developed to enable hackers to crack WEP and gain access to the WLAN.

To rectify WEP vulnerability, IEEE started to develop a more secure alternative named IEEE 802.11i [11] standard. However, the WLANs were already widely deployed, thus there was a need to have a stronger more secure alternative to WEP before IEEE 802.11i was released, therefore the WiFi Alliance [13] in conjunction with IEEE introduced an enhanced security scheme called WiFi Protected Access (WPA) [15]as an alternative to WEP in the first quarter of 2003.

WiFi 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, WiFi 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 and key distribution, IEEE 802.11i includes a negotiation process for selecting the correct confidentiality protocol and key distribution system for each traffic type. Other features introduced include key caching and pre-authentication. In this thesis we will only focus on WPA Enterprise which is described below.

2.2 WiFi protected Access (WPA)

In 2003, the Wi-Fi Alliance [13] introduced Wi-Fi Protected Access (WPA™)[15], which is a subset of the IEEE 802.11i specification. WPA replaces WEP with a comparatively strong encryption technology called Temporal Key Integrity Protocol (TKIP) with Message Integrity Check (MIC). It also provides a scheme of mutual authentication using either IEEE 802.1X/Extensible Authentication Protocol (EAP) authentication or pre-shared key (PSK) technology. WPA offers two modes of certifications each providing an encryption and authentication solution. Both modes are given below.

a) WPA Personal mode
b) WPA Enterprise mode

WPA Personal Mode utilizes TKIP for encryption and pre-shared key (PSK) as authentication mechanism. While WPA Enterprise mode makes use of TKIP for encryption and IEEE 802.1X/EAP as authentication mechanism. WPA2 also supports both personal and enterprise modes. Details of these both modes are given in table below.

Mode	**WPA**	**WPA2**
Enterprise Mode	Authentication: IEEE 802.1X/EAP Encryption: TKIP/MIC	Authentication: IEEE 802.1X/EAP Encryption: AES-CCMP
Personal Mode	Authentication: PSK Encryption: TKIP/MIC	Authentication: PSK Encryption: AES-CCMP

Table 2.1: WPA and WPA2 modes

2.2.1 Components of WPA Enterprise

2.2.1.1 Client Supplicant

An IEEE 802.1X supplicant is required on the client. A supplicant is software that is installed on the client to implement the IEEE 802.1X protocol framework and one or more EAP methods. Supplicants might be included in the client’s operating system, integrated into drivers, or installed as third-party standalone software.

2.2.1.2 Authenticator

The supplicant authenticates to 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. Usually an Access Point plays the role of authenticator.

2.2.1.3 Authentication Server

WPA-Enterprise employs IEEE 802.1X authentication with Extensible Authentication Protocol (EAP) types which provide mutual authentication on Wi-Fi networks. This helps to insure that only authorized users are granted access to the network and that users only access authorized subnets within the network. The requirements for an authentication server in a wireless network similar to those of a wired LAN; the authentication server stores the list of the names and credentials of authorized users against which the server verifies user authenticity. Typically, a Remote Authentication Dial-In User Service (RADIUS) server is used. User credentials may also be stored in an external database that can be accessed by the authentication server. The configuration is not defined in standards and can be implementation specific.

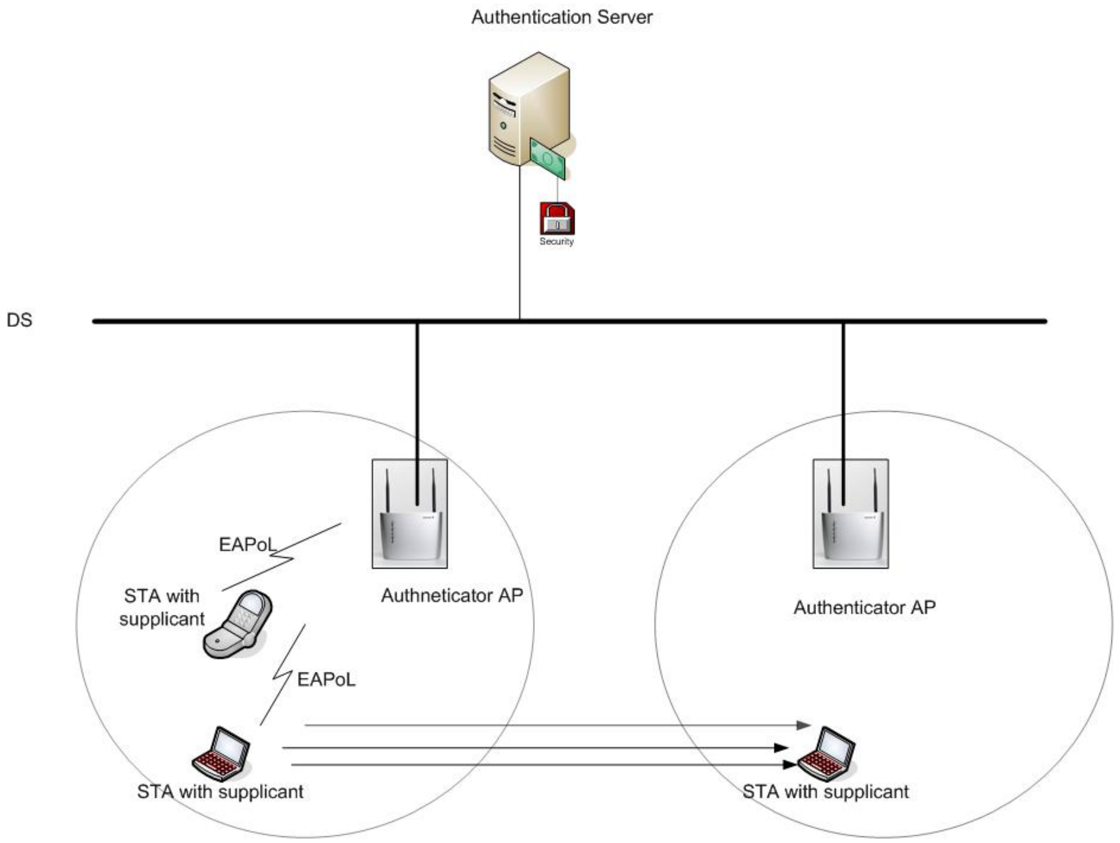


Figure 2.1: Components of WPA Enterprise WLAN

2.2.1.4 Extensible Authentication Protocol (EAP)Types

Extensible Authentication Protocol (EAP) types offer a range of options that can be used with different authentication mechanisms, operating systems, and back-end databases. Each maps to different types of user logins, credentials, and databases used in authentication. Possible EAP types include EAP-TLS, EAP-TTLS, PEAP, and other open standard types.

2.2.1.5 WiFi Protected Area Information Element (WPA IE)

In WPA enabled WLAN security parameters between Access Point (AP) and station (STA) are negotiated using beacon, probe response, and (re)association frames. The WPA enabled APs sends WPA IE in the beacon and probe response frames. This WPA IE contains information about security features and cipher suites supported by the AP. Based on its own security policy the STA selects security features and cipher suites from the APs WPA IE and constructs its own WPAIE which the STA sends in (re)association frames. This negotiation of security parameters is later validated during 4-way handshake.

2.2.1.6 Operational Framework

In WPA-Enterprise mutual authentication is initiated when a user associates with an access point. The AP blocks access to the network until the user can be authenticated. The user provides credentials which are communicated to the authentication server. The authentication process is enabled by the IEEE 802.1X/EAP framework. With EAP, IEEE 802.1X creates a framework in which client workstations and the authentication server mutually authenticate with one another via the AP. Mutual authentication helps to ensure that only authorized users can access the network and confirms that the client is authenticated to an authorized server.

If the authentication server accepts the user's credentials, the client joins the WLAN, otherwise the client remains blocked. Once the user has been authenticated, the authentication server and the client simultaneously generate a Pairwise Master Key (PMK), the 4-way handshake then takes place between the client and the AP to complete the process of authenticating the AP with the client, establishing and installing the TKIP encryption keys. As the client begins communicating on the WLAN, encryption protects the data exchanged between the client and the AP. [15]

2.2.2 Key Hierarchies

In the WPA Enterprise, an EAPoL-key exchange uses a number of keys and uses 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 figure 2.2. These keys are used in the EAPoL key exchanges. IEEE 802.1X defines an RC4 EAPoL-key frame. However, WPA defines its own EAPoL-key exchanges, *based on the IEEE 802.11i standard*. In the IEEE 802.11i specification, these exchanges are referred to as the 4-way handshake and the group key handshake.

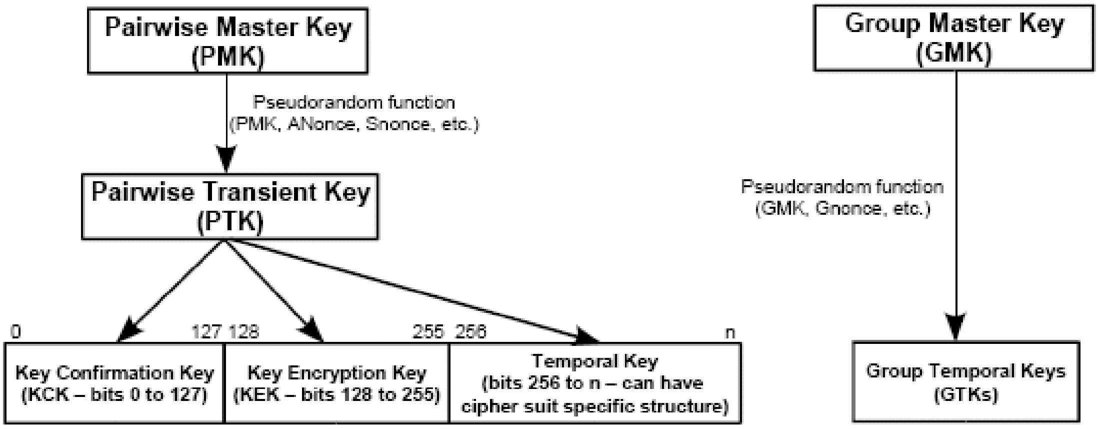

Pairwise Master Key (PMK)
Pseudorandom function (PMK, ANonce, Snonce, etc.)
Pairwise Transient Key (PTK)
0
127 128
255 256
n
Key Confirmation Key (KCK – bits 0 to 127)
Key Encryption Key (KEK – bits 128 to 255)
Temporal Key (bits 256 to n – can have cipher suit specific structure)
Group Master Key (GMK)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

Figure 2.2 : Pairwise Master Key (PMK) and Group Key hierarchy [21]

2.3 Azimuth Systems

The Azimuth W-Series WLAN Test Platform [26] is a wireless test network that allows to perform automated, sophisticated and advanced testing and measurements of 802.11 wireless devices that result in repeatable, reliable, and consistent test results. This Test Platform allows creating an 802.11 simulated test environment in which attenuation is used to virtually distance test devices from each other. By attenuating signals in the Azimuth W-Series WLAN Test Platform, we can virtually move stations, access points (APs), and application specific devices (ASDs) closer or further away from each other. By using attenuation to virtually position devices, and by customizing traffic using the internal traffic generator, we can create a wide variety of test scenarios such as roaming stations, overlapping/non-overlapping BSSs, and hidden stations.

Internal traffic generators in the Azimuth Systems solution enable us to determine the origin and destination of traffic and customize frame patterns, frame lengths, numbers of frames, and test duration. The Azimuth W-Series WLAN Test Platform includes the following major components

2.3.1 Azimuth 801W/800W

This RF-isolated chassis in the Azimuth W-Series WLAN Test Platform houses eight modules that are used for testing and measurement of 802.11 wireless devices. The eight front-loading modules house up to 16 stations in individual RF-isolated chambers and provide a variety of functionality including emulating hundreds of clients, emulating variable distances between clients and APs, and providing traffic generation capabilities. Modules for the Azimuth 801W/800W Chassis are given below

2.3.1.1 Station Test Module (STM)

It houses two PCMCIA, mini-PCI, USB or CardBus wireless RF network identity card NICs in separate RF-isolated chambers.

2.3.1.2 Wireless LAN Analyzer (WLA)

It houses two built-in stations that each run WildPackets AiroPeek 802.11 Wireless LAN Protocol Analyzer software [25].

2.3.1.3 TestMAC Module (TMM)

The TestMac Module (TMM) emulates *from one to hundreds of stations* (softClients), each with its own MAC address. A traffic generator allows all softClients to send and receive traffic. This module is
especially useful in system loading and stress testing.

2.3.1.4 RF Port Module (RFM)

This module is used to connect 802.11 stations, APs, ASDs, and softClients to create sophisticated test scenarios including roaming stations, overlapping/non-overlapping basic service sets (BSSs), and hidden stations.

2.3.1.5 RF Test Heads

Azimuth offers two different types of test heads for housing access points (APs) or application specific devices (ASDs)

- **Azimuth Mini RF Test Head (MTH)**
  It houses multiple APs and ASDs for testing and measuring 802.11 devices in the Azimuth W-Series WLAN Test Platform. The compact MTH consists of two RF-isolated chambers that provide greater than 90 dB of isolation that prevents unwanted radio frequency interference (RFI) from either entering or exiting the chamber.

- **Azimuth Laptop RF Test Head (LTH)**
  It houses a laptop for use in testing and measuring 802.11 devices in the Azimuth W-Series WLAN Test Platform. The Azimuth LTH is an RF-isolated chamber that provides greater than 110 dB of radiated RF isolation between the device under test and the outside world. RF isolation prevents unwanted radio frequency interference (RFI) from either entering or exiting the chamber.

2.3.1.6 Azimuth DIRECTOR

This application is the command and control center of the Azimuth W-Series WLAN Test Platform. The Azimuth DIRECTOR consists of a PC and software application that communicates with the system over three separate and distinct Ethernet LAN connections to perform various tests, gather statistics, run bench mark applications and communicate with outside networks.

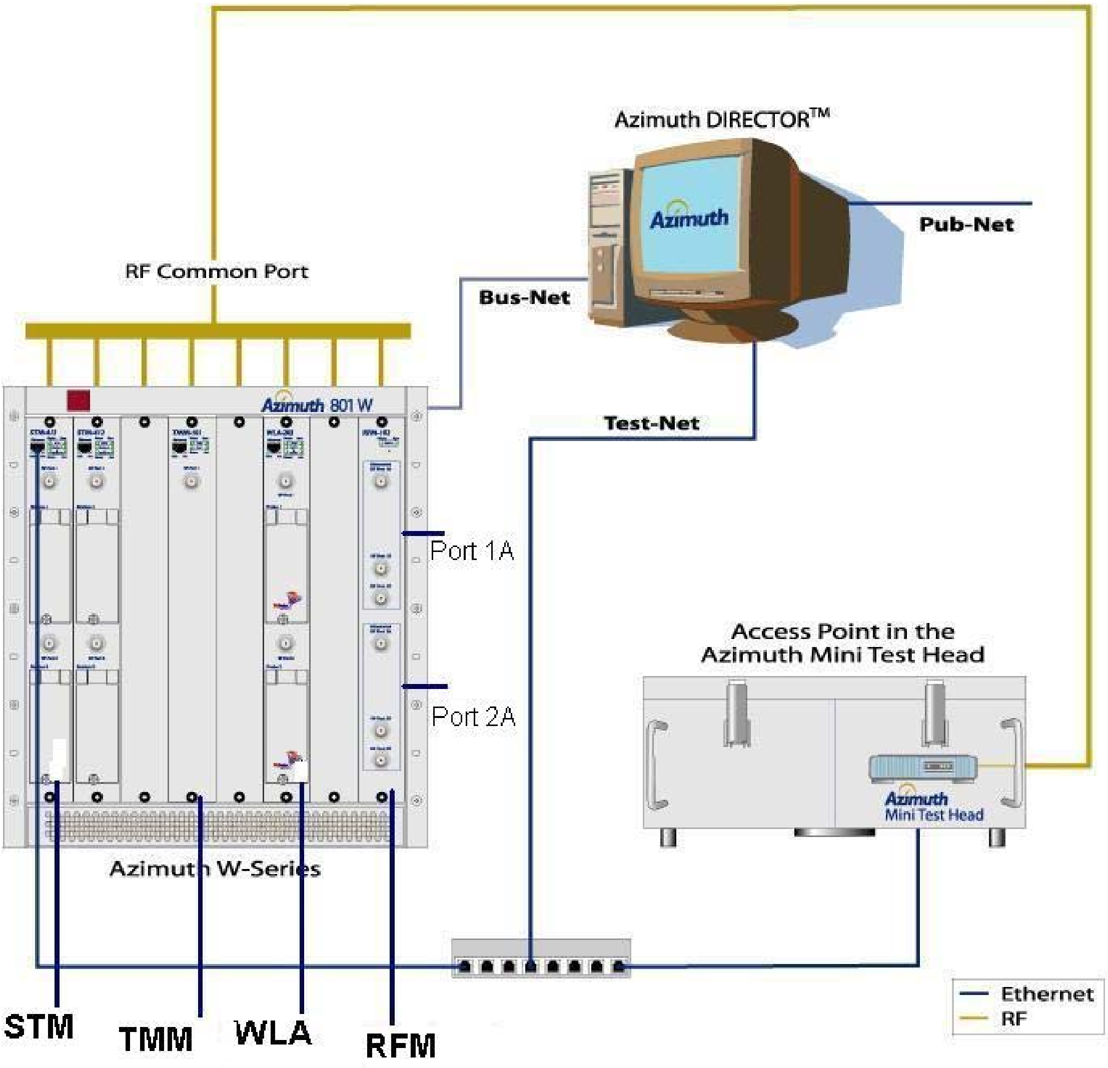


Figure 2.3: Azimuth Systems [26] [1]

[1] Original picture has been taken from Azimuth Systems User Manual .

2.3.2 Smooth Roaming

The Smooth Roaming Benchmark application uses the RF port Module (RFM) to force a station to roam between two APs. The AP that the station associates with before roaming is considered the *origin AP*. The AP that the station associates with after roaming is considered the *target AP*

There are RF attenuators on the RF connections to these two access points. The RF attenuators are controlled so as to decrease the signal strength of one access point and increase the signal strength of the other - thus emulating a handoff. Smooth Roaming Benchmark application can perform a detailed analysis and measurement of the smooth roaming behavior. During the smooth roaming tests the RF port Module (RFM) is connected to two APs, a single client card *(under test)* is placed in Station Test Module (STM) and a WLAN Analyzer (WLA-202) is set to analyze the traffic between the devices (Please see figure 2.3).

The Smooth Roaming Benchmark application adjusts the RFM attenuators on Port 1A and Port 2A (the attenuated ports) to force roaming between the two APs. By increasing and decreasing the attenuation, the station is virtually positioned closer or further away from an AP. Decreasing the attenuation virtually positions the station closer to an AP. Likewise, increasing the attenuation virtually positions the station further from an AP. By adjusting the attenuators in these ways, the Smooth Roaming Benchmark application forces roaming between the two APs. *Theory of operation* of smooth roaming bench mark application is illustrated in Fig 2.4 while figure 2.3 illustrates the interconnection between various modules when smooth roaming test was performed. Further details about the Azimuth systems and smooth roaming bench mark applications could be seen in Azimuth user documentation [25].

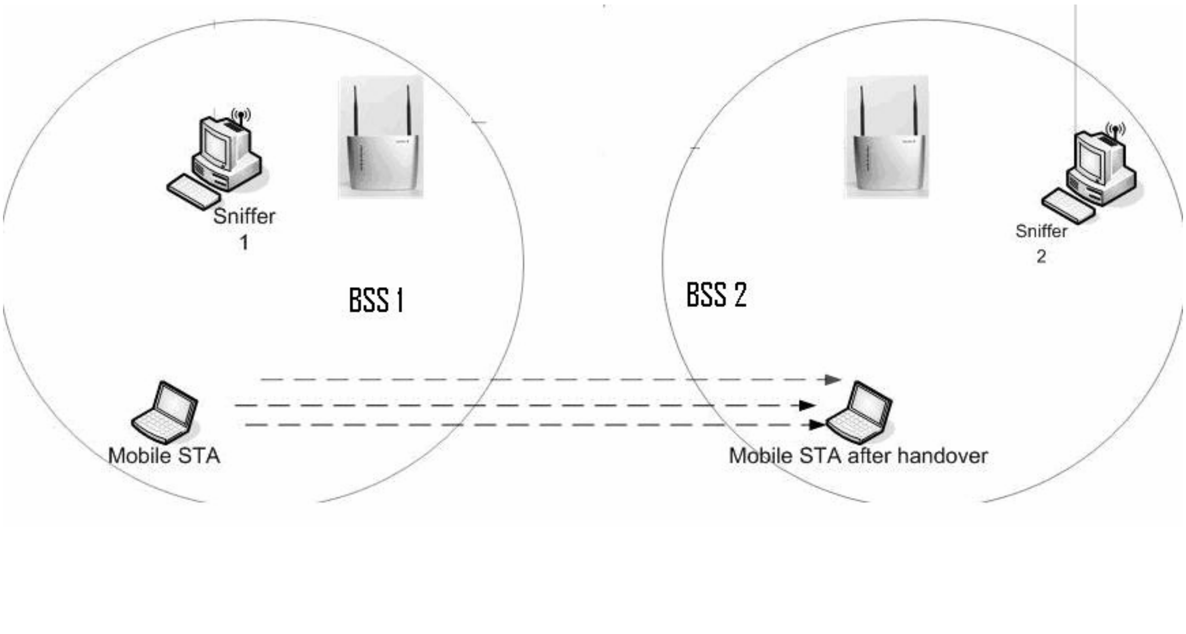


Figure 2.4 . Smooth Roaming bench mark application- theory of operation

2.4 Earlier Work

Given the potential market size of WiFi based Voice over IP service (VoWiFi); there has been lots of research concerning VoWiFi roaming. Researchers have carefully considered the factors affecting handover delays through both simulations and physical tests. The following is a list of some of the relevant prior work:

- Jon Olov Vatn's doctoral dissertation "IP telephony: mobility and Security"[12] deeply analyzed the possible factors which might effect the handover time. He has considered open authentication and the calculated the associated handover delays. He gives some suggestions of how to reduce the handover delays in an 802.11i based VoWlan (or VoWiFi) by performing the Pre-authentication with the new Access point (AP) while still connected via the old AP.

- Ajeet Nankani analyzed the impact of the EAP-TLS authentication system on voice like traffic for WLAN handovers [21]. In his thesis Ajeet advanced some of the work done by J.O. Vatn by performing 802.11i based handover tests.

- H. Velayos and G. Karlsson also studied handovers. They analyzed the link-layer handoff process in WLAN based on the IEEE 802.11b standard and made some suggestions about how to reduce its duration [24]. However, there are some questions about their method of triggering handoffs (i.e., as they simply powered off one of the APs).

In this thesis we performed tests in real networks to measure handover delays for WPA Enterprise mode. We performed detailed analysis of each higher layer authentication phases. We used **actuall voice traffic** and analysed the impact of handover on various VoIP clients. Post higher layer authentication phases of the STA and impact of the various internal modules of Access Point on overall handover latency have been studied in this thesis.

3 Handovers

When a mobile STA is operating in *infrastructure* mode it tries to *associate* with an AP in its vicinity. Each AP constitutes a *basic service set* (BSS) and all the traffic to and from the STA will go via this AP, even traffic between two STAs associated with the same AP. To cover a larger area, multiple APs can be connected via a *distribution system* (DS) to form an *extended service set* (ESS). A STA moving out of the coverage area (cell) of one AP could *reassociate* with another AP (within the same ESS) in the new location, thus performing a layer-2 (link layer) handover. APs with overlapping coverage areas are commonly configured to operate on different frequency channels to avoid interference between the cells.

The standard does not specify the design of the DS [16], but a commonly used solution is to connect a set of *bridging APs* via one or several Ethernet bridges as shown in Fig. 3.1.When a STA moves away from its original AP, the signal strength of the messages received from that AP will decrease. At a certain signal strength threshold, the STA will start to look for a better AP to associate with, if it finds one it will trigger a handover. The standard specifies that a STA can only be associated with one AP at a time [16], so there is a risk that communication is interrupted while the STA performs the handover. The duration of the period when the STA in unable to exchange data traffic via its old *Origin* or new *Target AP* is often referred to as the handover latency or handover delay. However, the precise definition of handover is more complicated [12] .

We studied handovers in a scenario as presented in figure 3.1. One major difference between our work and most previous work is the use of real voice traffic rather then traffic generators and the use of the Azimuth system to emulate the RF environment. We realized that utilizing traffic generator for emulating voice traffic can provide the data for statistical analysis with respect to number of packets lost, but it ignores an important aspect in understanding actual behavior of VoWiFi handovers. Details of this phenomenon are discussed under the section 4.3. By using an actual AP rather then Linux HostAP we did observed that the wrong authentication phase never appeared in our case (observed by both Vatn and Ajeet). However we are not sure whether use of the same chipset cards or Linux HostAP was the reason for wrong authentication phase observed by both Vatn [12] and Ajeet [21]. We confirmed our results by matching them with automated tests form of Azimuth environment.

3.1 Testbeds

We used three different methods to elucidate our results.
1. Testbed using actual WLAN clients with commercial Access point.
2. An emulated RF environment using Azimuth Systems (described under section 2.3)
3.Pre-Configured scripts for testing authentication time

The first method was primarily used to measure the EAP-TLS [29] based authentication delay for logging *into Wireless Protected Access (WPA) enabled WLAN*. This method was also used to observe Post EAP-TLS [29] authentication behaviour as well as the impact of handover on various VoIP user agent clients (UAC). Additionally we also performed end-to-end handover tests with this method. Limitations of this method could be seen in 3.2.1.1.

The second method was primarily used to measure total handover delay for open authentication. Although WPA based handovers were also measured in this method.

The third method was used only to verify the EAP-TLS based authentication delay, where we made use of *Pre-Configured* scripts *written in the Ericsson Customer Premises Equipment Products and Services (CPEPS) testing Lab* for functional testing. These scripts were written only to test various performance parameters of Access Point itself rather then to roaming delay. For roaming measurements we mainly relied on a real system testbed and Azimuth Systems.

3.1.1 Testbed using Real System

We have mainly examined **Intra** ESS handovers, i.e., where all the APs are configured to belong to the same extended service set (ESS). It was also assumed that the STA performs *active* scanning when searching for candidate APs. As the IEEE 802.11 shared key authentication mechanism wired (WEP) is **not** utilized for access control, i.e., open system authentication is used. We performed roaming tests for Wireless Protected Area (WPA) specific scenario where EAP-TLS [29] was used as the preferred EAP method for authentication. Our STA was only equipped with a single WLAN interface. Although we also tried some Inter ESS handovers (limiting ourselves to IPv4 based handovers), however the results of such handovers are not the primary focus of this report. It is worth mentioning that handoff was triggered **after** establishing the VoIP call and there was no other traffic going on during the tests.

Further details about the testbed components can be found in Table 3.1 given below. The reason for using this testbed was to closely observe the behavior of a VoWiFi system , by observing roaming phases in detail, and to identify key aspects which were not explicitly defined (but were available) in standard testing scripts or tools for the second test method. We found this phase very useful in revealing the behavior of different *voice clients and services* (see chapter 4).

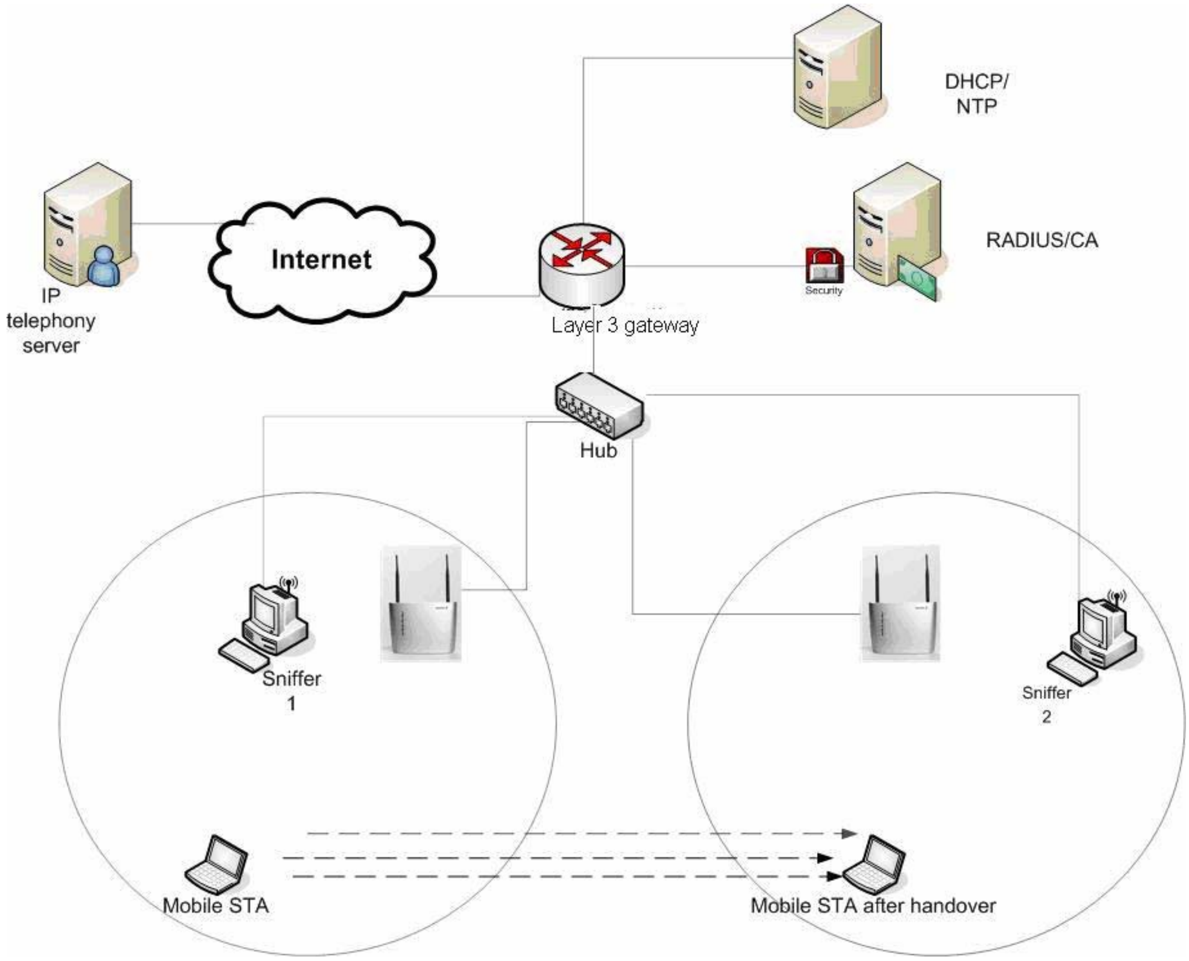


Figure 3.1: Roaming scenario diagram in the real system testbed ; a Mobile STA moves from its Origin AP → Target AP

Real System Testbed Components			
Item	**Description**		
Analyser hardware	**Client STA**	**Sniffer 1**	**Sniffer**
WLAN cards	Netgear WG 511T 108 mbps wireless PC card	DLink-AirXpert 802.11 ABG WLAN PCI card WiFi Analyser 1	Netgear WAG 511 Dual band wireless PC card (Wifi Analyser 2
Computers/OS	HP Compaq nc6000 P 4 , Windows 2000	DELL Optiplex GX1 PIII Windows XP SP2	HP omnibook PIII Windows XP SP2
Ethernet cards		3 Com 3c918 Integrated fast Ethernet controller	3com 10/100 mini PCI Adaptor
Analyser software Ethernet		Ethereal V 0.10.11 Winpcap V 3.0 Windows XP SP2	OpenXtra Ethereal V 0.10.12 Winpcap 3.1
Analyser software (WiFi) / client utility	Odyssey client manager Version 4.04.0.2112 Funk software Client utility	Commview V 5.0 WiFi analyser	Commview V 5.0 WiFi analyser
RADIUS Server *Access points*	Funk RADIUS Ericsson ABS 2200	*Authentication method*	EAP-TLS

Table 3.1 : Real system testbed components

3.1.2 Traffic for Testing (Real System Testbed)

We have generated voice traffic by registering with various SIP based (and propriety voice) services and then maintaining a real time RTP session between a *caller* and *callee*. We selected one of several prerecorded **speech** samples from a reverse speech website [28] which was played in auto repeat mode at one end node; while at other end a user spoke during silent periods (if required).

3.1.3 Handoff Triggering

Both the origin and target AP’s were reasonably near to each other and I walked carrying a laptop (equipped with WLAN card) to cause a handover between *origin AP* was configured with lower output power and without an antenna while the *target AP* was configured with comparatively higher output power and with antenna. Initially the laptop STA was placed very near to the *origin AP* until it was connected to the origin AP, then I started moving towards the target AP which was already set with higher output power. To avoid interference (to a limited extent only) we configured our mobile client with a WPA based profile which forced our STA to only connect to WPA capable AP’s , and hence it did not attempted to connect to other AP’s in the surrounding area which were using **open** authentication. However, it still had to listen to these other AP’s and their frames as well as respect the CSMA/CA MAC protocol - thus the timing of access to the network was affected by these other networks. No attempt was made to quantify this effect, however, the emulation environment does not suffer from this interference and hence can be used for controlling this. It is worth mentioning that handoff was triggered **after** establishing the VoIP call. There was no other traffic *going on* during the tests.

3.1.4 Processing delay / Idle Time

Time spent between two phases is either utilized in processing the information retrieved from the previous phase, by the AP or the STA or remaining idle. By examining order of messages to be exchanged between an AP and a STA as per particular authentication method or standard used, we can guess whether the AP or STA was responsible for the particular processing delay/ Idle time period These delays constitutes a significant proportion of total hand off latency other then the time spent over the air and hence propose the improvements required in a particular module at firmware or hardware level. We paid special attention to such delays while calculating overall handover latency. In addition, we used knowledge of some of the internal APIs to improve our "guess" as to who was responsible for specific parts of the delay. Regarding processing delays / idle time spent at STA, unfortunately we couldn’t categorize that wither this time was taken by various processes at firmware layer or at operating system layer.

3.2 Handover Phases

In handovers a STA moves from an origin AP to new target AP. To do so, a STA continually monitors the observed link quality from its current AP and uses this information for 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 [16], so it may be as simple as signal to noise ratio measurements or may combine many other parameters from the entire WLAN system including received signal strength indicator (RSSI), frame success rate (FSR), bit error rate, packet loss. To simplify the *Intra ESS handoff analysis*, handoff will be divided in following phases described in subsequent sub section.

3.2.1 Continuous Unacknowledgment Phase

This phase is defined as phase observed from after the last acknowledged packet sent to or from the *origin AP* until first probe request received at the *target AP*.

The duration of this phase is depends on various factors, including the coverage of the wireless cell, movement of the mobile STA, fading, the load on the target AP etc. However, the most important factor is the **handoff triggering criteria used by STA** , standard [16] doesn't describe the criteria for handovers triggering, so it is dependant on vendor specific implementation which could be used to address a particular target market. With respect to criteria used for *handoff threshold* I divide STA's into two main categories.

a) Network / Hotspot friendly STAs
b) Single AP /Home user friendly STAs

Network friendly STA's are more aggressive in proceeding to the next phase i.e. scanning phase as they quickly decide that the *Origin AP* is no longer available or atleast not *worth waiting* for any longer and hence they have a comparatively very short *continuous unacknowledgment phase*. On the other hand single AP /home user friendly STA take comparatively a very long time before they begin scanning new AP. But we believe that that **only** *handoff threshold criterion* **shouldn't** be the only factor in rating STA as *hotspot* or *home* friendly.

Packet retransmission can occur quite often in a WLAN and it is very common phenomena. This can occur due to collisions, packet loss, multi path fading or user being out of range etc. We also have observed that sometimes a few packets are unacknowledged but they are soon followed by transmission of normal acknowledged packets. So counting **from** the first unacknowledged packet requires careful observation and careful and sophisticated implementation of testbed. We believe that in order to until measure precisely this phase should be measured from "after the last acknowledged packet until the reception of first probe request". This is the **most important** and **critical** phase in the overall handover process where the maximum packet loss is likely to occur.

3.2.1.1 Observations

We observed that our first WiFi analyzer (CommView) was occasionally missing packets while capturing frames, especially Control/ACK packets. This made it difficult for us to accurately identify the first unacknowledged packet in our first Testbed. However, with the help of our second testbed , i.e., *Azimuth Systems* emulation environment ,we were able to accurately measure the *range* of duration for this phase.

We have observed that at a certain *path loss difference* between **Target AP** and **Origin AP** the STA detects that the mobile has roamed. Although this path loss difference is not fixed but normally at a path loss difference of 16-20 db the STA detects roamed (*handoff threshold*) and it starts scanning for a candidate APs. However, we have also noticed STA detecting the roam (*handoff threshold*) at a path loss difference of 2 db, occasionally. As path loss difference is the outcome of various other WLAN parameters, so this itself might not be the only decisive handoff threshold. Its worth mentioning that we noticed that time required for *detecting roaming* based on a 16-20 db path loss difference, varies over a **very** wide range we will address some possible sources for this variance in section 3.4.6.1.

3.2.2 Scanning Phase

To find candidate APs to associate with the STA will *scan* the different radio channels; scanning can either be done passively (by listening for beacon messages from APs) or actively by sending a *probe request* message on each channel and listening on that channel for *probe responses* from APs, The scanning phase begins by scanning for APs. The STA must wait for *probe_delay_time* before starting the scanning process, which can be either passive or active. Here we will examine the scanning phase when scanning is done actively.

1. Scan phase starts, after starting probe delay timer, the *current_channel* is set to 0.
2. STA waits until Probe delay timer reaches *probe_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 or STA does not see any traffic, then the *current channel* is assumed empty and the STA goes to step 3 to start same process for the next channel, otherwise if a probe response or traffic was seen on the *current channel* then the 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 to perform the same process for next channel.
8. Once all channels have been scanned it sorts out the processed scan results and picks the best APs, i.e., those which may provide the best link quality and have the matching WPA IE.
9. Scanning phase ends.

In Scanning phase the STA will send at least one probe request and may receive zero or more responses per channel, depending on the number of APs on that channel serving the ESS specified in the probe request (please see figure 3.2). It is assumed that active scanning is used. [21]

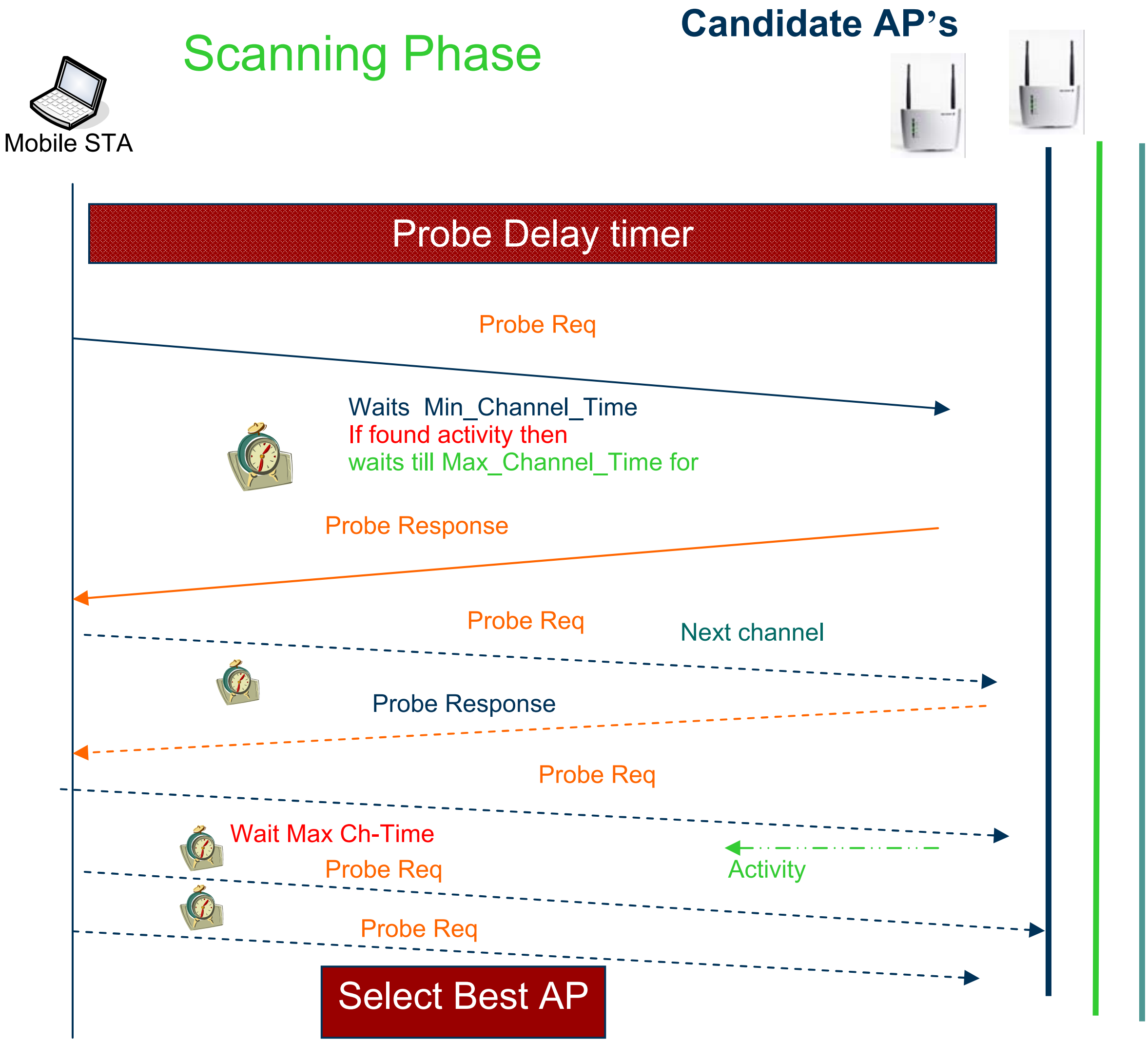


Figure 3.2 : Scanning process

Scanning behavior of STA's can be classified into atleast two main categories.

a) Network/Hotspot friendly STAs
b) Single AP /Home user friendly STAs

A *Hotspot friendly* STA will scan the radio channels in method described above , where the *next channel* could be *sequentially next channel* or one selected through *selective scanning mechanism* as described by Shin[22] or similar. After scanning the channels only once, they aggressively proceed to the next phase.

Home user friendly STAs are reluctant to hand over to new AP and try to remain associated with current AP as long as possible. They might even use *multiple rounds of scanning* even when they find a new AP at reasonably good signal strength. Such STA's rather then proceed to the next phase immediately after scanning all the channels once, tend to scan the channels multiple times.

However we believe that a STA **shouldn't be** rated *Home friendly* or *Hotspot friendly* on the basis of scanning behavior **alone**. A STA might exhibit network friendly behavior in scanning phase, but might not necessarily show *network friendly* behavior in rest of the other phases.

3.2.3 MAC Layer Authentication phase

When the STA has finished scanning for candidate APs, it will initiate the association procedure with the best target AP if such an AP exists. But before associating with the desired AP it goes through an authentication phase, this phase utilizes the BSSID (MAC address) of best AP learned during the scanning phase (as described in section 3.2.2) and tries to connect to that AP by sending an authentication message to the selected AP, hoping to retrieve a success message in an authentication response frame from this AP. Authentication phases can use *wired equivalent protection* (WEP) encryption or *open system* as the method of layer 2 authentication.

3.2.4 Association Phase

Once authenticated the STA sends an association frame and expects a association response frame from this AP. This indicates that the STA is now associated with this new AP. In our case this association is temporary as the STA still has to pass through a higher layer authentication phase for permanent association.

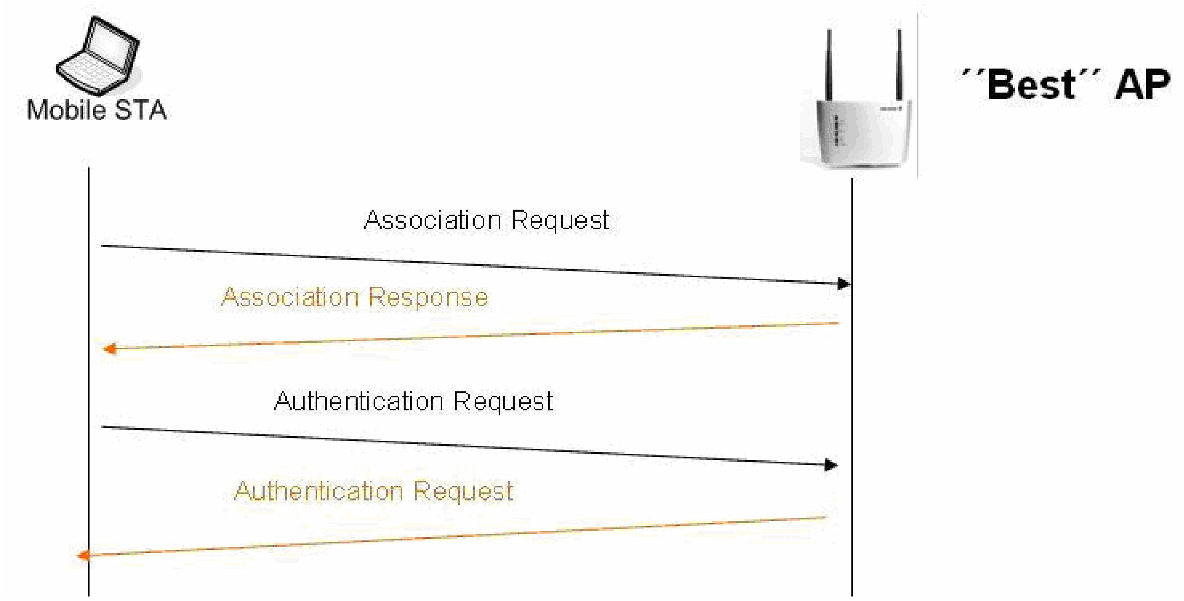


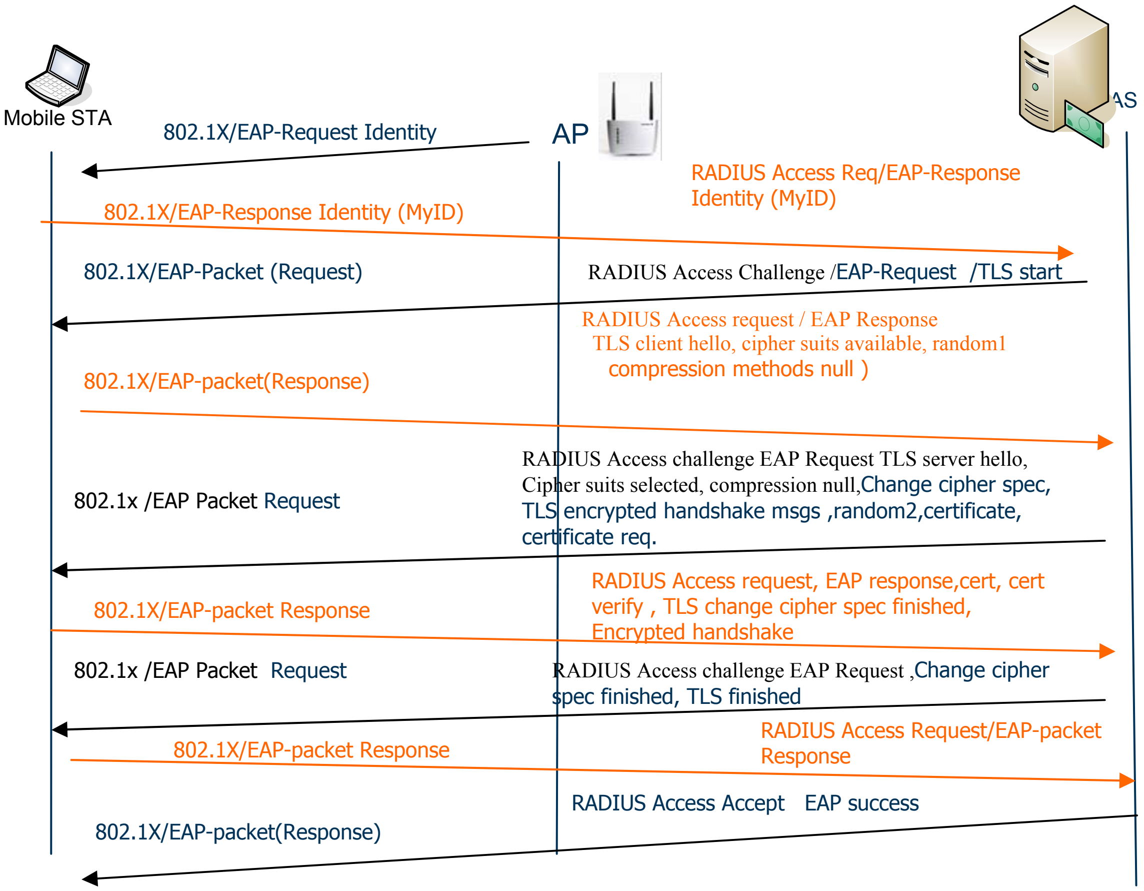
Figure 3.3 : Association and Authentication Phase

3.2.5 Higher layer Authentication phases

In the case of WPA enabled authentication utilizing EAP TLS as the EAP method, there are three sub-phases of higher layer authentication. Three phases are defined below.

3.2.5.1 PMK derivation phase

In this phase both the Authentication Server (AS) and the STA derive the Pairwise Master Key (PMK) by exchanging multiple Radius Access / Challenge messages. In this phase the certificate keys are exchanged and verified to allow **mutual** authentication of STA and AS. Random numbers from both sides are also exchanged in this phase , as a result of this authentication and exchange of random numbers both STA and AS are each able to derive a Pairwise Master Key (PMK) which is used for generation of Pairwise Transient key (PTK) in next phase. Usually this phase starts with a EAP identity request from AP → STA and ends upon receiving a RADIUS Accept packet AS → STA.


Mobile STA
AP
AS
802.1X/EAP-Request Identity
802.1X/EAP-Response Identity (MyID)
RADIUS Access Req/EAP-Response Identity (MyID)
802.1X/EAP-Packet (Request)
RADIUS Access Challenge /EAP-Request /TLS start
RADIUS Access request / EAP Response TLS client hello, cipher suits available, random1 compression methods null)
802.1X/EAP-packet(Response)
RADIUS Access challenge EAP Request TLS server hello, Cipher suits selected, compression null,Change cipher spec, TLS encrypted handshake msgs ,random2,certificate, certificate req.
802.1x /EAP Packet Request
RADIUS Access request, EAP response,cert, cert verify , TLS change cipher spec finished, Encrypted handshake
802.1X/EAP-packet Response
802.1x /EAP Packet Request
RADIUS Access challenge EAP Request ,Change cipher spec finished, TLS finished
802.1X/EAP-packet Response
RADIUS Access Request/EAP-packet Response
RADIUS Access Accept EAP success
802.1X/EAP-packet(Response)

Derive PMK : TLS PRF (Master key, client EAP Encryption , random1 ,random2)

Figure : 3.4 PMK Derivation Phase

3.2.5.2 PTK Derivation (4-way handshake) Phase

After the derivation of a PMK, the AS pushes the PMK to the AP , now the AP can generates the PTK with the STA. In this phase both AP and STA exchange Nonces. A Pseudo Random Function (PRF) takes the Nonce and MAC of both AP and STA as input to derive PTK. This 4-way handshake phase starts by sending a Nonce AP→STA in a EAPoL key message and ends with a unicast EAPoL key message STA→ AP. At the end of this phase both STA and AP have Temporal Keys.

3.2.5.3 GTK Distribution Phase

In this phase the AP sends Group Transient key (GTK) to the STA by encrypting it with a Key Encryption Key (KEK) and authenticates it with Key Confirmation Key (KCK). Both the KCK and the KEK are generated using the PTK. In this phase only 2 messages are exchanged.

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 departing STA from receiving any new multicast or broadcast messages from the AP.

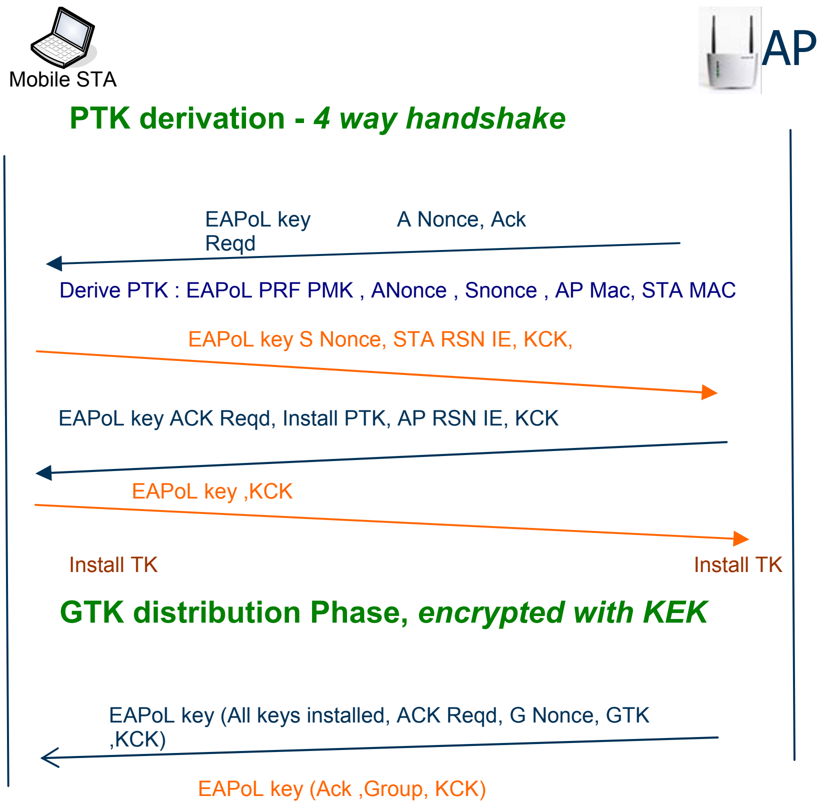

Mobile STA
AP
PTK derivation - 4 way handshake
EAPoL key Reqd
A Nonce, Ack
Derive PTK : EAPoL PRF PMK , ANonce , Snonce , AP Mac, STA MAC
EAPoL key S Nonce, STA RSN IE, KCK,
EAPoL key ACK Reqd, Install PTK, AP RSN IE, KCK
EAPoL key ,KCK
Install TK
Install TK
GTK distribution Phase, encrypted with KEK
EAPoL key (All keys installed, ACK Reqd, G Nonce, GTK ,KCK)
EAPoL key (Ack ,Group, KCK)

Figure 3.5 : PTK Derivation and GTK distribution Phases

3.2.6 Post Higher Layer Authentication Phases

During Post Higher Layer authentication three possible behaviors could be seen. These three cases:

Case 1 – DHCP and Gratuitous ARP Phase
Case 2 – ICMP Exchange Phase

Case 3 – Ongoing voice traffic on uplink updating the distribution side (DS)

3.2.6.1 CASE 1

3.2.6.1.1 DHCP Phase

In this phase the STA requests an IP address and is assigned one by DHCP server. The assigned IP address is same which STA had during association with previous AP. This phase typically consists of 2 messages; DHCP Request and a DHCP response with the assigned address. However, its worth mentioning that intra ESS handovers are performed within same subnet so *theoretically* DHCP phase shouldn't even occur. Possible explanation for this behavior is described under section 3.3.1.7.1.

3.2.6.1.2 Gratuitous ARP

After assignment of an IP address the STA may send a Gratuitous ARP broadcast message to update the station cache on due to changes in forwarding table with in of the AP. This behavior is different from the one mentioned in 802.11F and it helps to quickly update the station cache of the Distribution System.

The IEEE 802.11F standard [30] specifies the use of a *link-layer update* message to update stale station caches in switches of a layer-2 distribution system upon a handover. The link-layer update frame is a link layer-broadcast message sent by the new AP to the distribution system on behalf of the mobile station, i.e., the link-layer update frame will use the MAC address of the mobile station as its source address**.**

NB *: Its worth mentioning that our AP's were **not** using 802.11F.

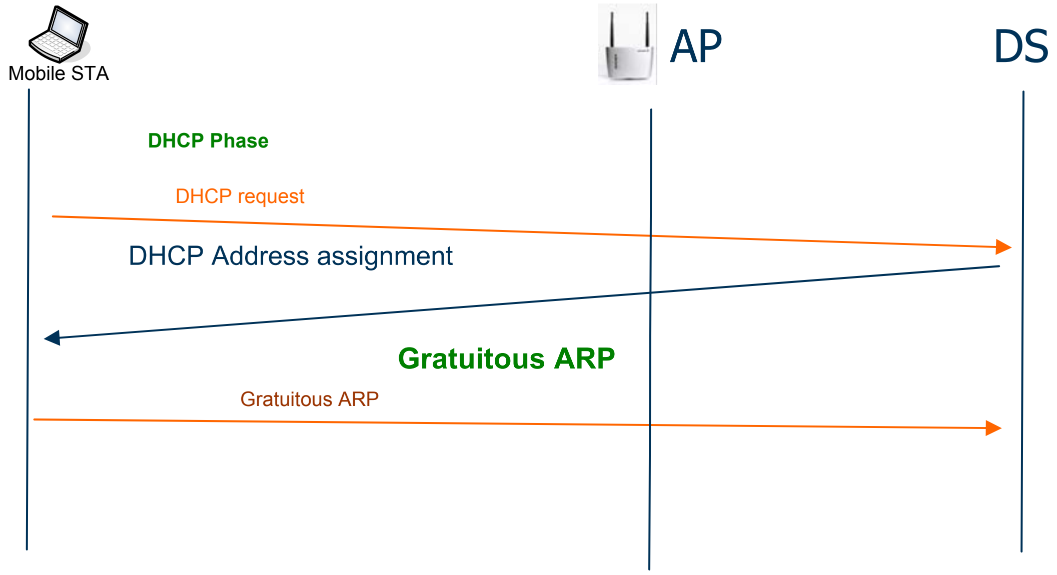


Figure 3.6 : DHCP-Gratuitous ARP Phases (Case 1)

3.2.6.2 CASE 2

3.2.6.2.1 ICMP Exchange Phase

In the ICMP Exchange phase, the STA sends an ICMP Request to its layer 3 gateway on the distribution side using the same IP which it held in associated with the previous AP and in turn it receives an ICMP Response from this layer 3 gateway , hence updating the station cache of the distribution side. This phase may appear when there is no other ongoing traffic left between the STA and the Distribution side.

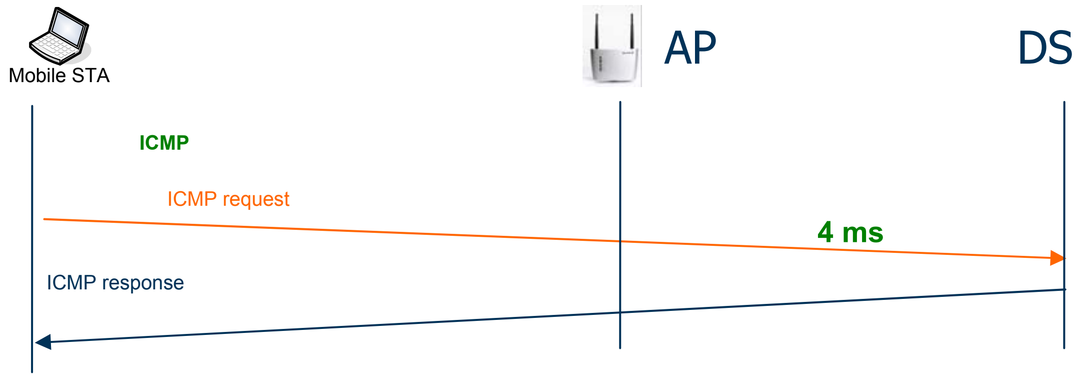


Figure 3.7 : ICMP Exchange Phase (Case 2)

3.2.6.3 CASE 3

3.2.6.3.1 Ongoing Voice traffic on Uplink

If there is ongoing voice/data traffic on uplink from STA towards the distribution side then a voice packet from STA can quickly update the station cache on the distribution side.

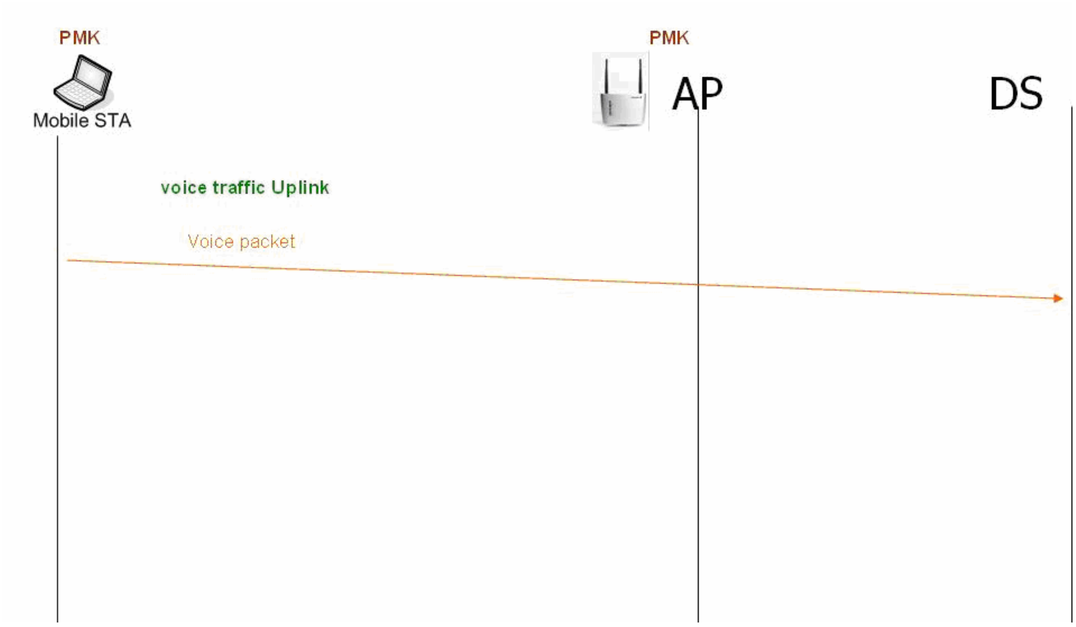


Figure 3.8 : Ongoing voice traffic (Case 3)

3.3 Details of Testing

- In *real system testbed* we located our laptop very near the desired AP before starting each measurement session. After starting both sniffers , we plugged in the PCMCIA WLAN which implements an (STA) thus the sniffer could capture packets starting from the first probe request until the last probe response to calculate the duration of scanning phase.
- We have paid special attention to the measurement of the *Processing Delay/Idle Time*. *Processing delay / Idle time* is the time when the STA and/or (more frequently) the AP was either processing the information it retrieved in previous phase or was idle. This has given us good insight into measuring the fraction of time spent versus *Processing delay or remaining idle* out of the total handover delay. We have compared the amount of time spent exchanging various packets *over the air* and *Processing Delay/Idle Time* of various phases of entire handover process. The results helped in highlighting that improvements required at hardware or firmware level in reducing handoff latency .We noticed *nearly the constant* **gap** *(Processing Delay/Idle Time)* between some phases and attributed that time to possible processing activity during that period.
- We did 40 experiments *with the real system testbed* and 200 experiments with the help of the *Azimuth systems emulation environment*. In real system test bed we measured all the phases which are described in section 3.2 .However, after making all of the 240 measurements in final output we preferred results obtained from the Azimuth systems for both the *continuous unacknowledgment phase* and the *scanning phase* due to fact that CommView (WiFi sniffer) occasionally missed frames, which made it difficult for us to *especially* measure the duration of these two phases.

- Its worth mentioning that the Azimuth systems *smooth roaming benchmark application* doesn't divide the overall output in to the phases we mentioned under section 3.2.
- Results obtained for Higher layer authentication and Post Higher Layer authentication phases from both methods were more or less similar and nearly constant. After reaching this conclusion we used Azimuth largely for measuring handover delays for *open authentication*. Our main concern in utilizing *Azimuth systems* was to verify *strange* behavior of different STA's in continuous unacknowledgment phase and scanning phase.
- Azimuth system doesn't divides its output in to the phases mentioned in section 3.2. we utilized *Smooth roaming bench mark application* of to verifying our results (see section 2.3)
- During PMK derivation the STA and the RADIUS server exchanged 20 messages including 8 messages (4 pair of Challenge/Response exchanges) to exchange the 4710 bytes of server certificate, which was divided in 4 EAP fragments of 1286, 1290,1290, & 844 bytes while 6 messages (3 pairs of Request/ Response) were to exchange a client certificate of 2863 bytes in 3 fragments of 1389, 1394, and 79 bytes. These large certificates result long delays for Pairwise Master Key (PMK) derivation phase. On the client side the RSA Public key length of 1024 bits was used with SHA1RSA signature algorithm of .Theoretically when the certificate sizes are below the fragmentation threshold, only 8 messages should have been exchanged between STA and RADIUS.
- We have used three different vendor STA hardware in Netgear , LinkSys, and cisco but main work was done using Netgear STA.
- Management frames including Beacons , Probe Request and response were sent on 1Mb/s. The Control/ACK frames were generally also sent at 1 Mb/s, but many times we also observed Control/ACK frames sent at 24 Mb/s. However, data frames were sent at higher data rates of up to 54 Mb/s, while retransmission of frames commonly slows down to lower (possible) data rates in a higher probability of a successful transmission.

3.4 Measurements of Handover Phases

We measured delays for all the handover phases described in section 3.2 and their associated *processing delays / Idle time*. The results obtained for Netgear WG511T over 100 tests are given below.

Step	Handover Phases	Duration (ms)			
		Sub group	Min	Avg	Max
1-	**Continuous Unacknowledgment**		11	387	1648
2-	**Scanning**		57	1112	1520
	Processing delay/Idle time		0	*1*	*65*
3-	**MAC Authentication**		5	38	50
	Processing delay/Idle time		0	*1*	*11*
4-	**Association**		1	28	52
	*Processing delay/Idle time**		*158*	*158*	*158*
5-	**Higher Layer Secure Authentication**		298	416	462
	Processing delay/Idle time		*81*	*179*	*491*
5a	PMK derivation (wifi) **	413			
	Processing delay/Idle time	16			
5b	PTK Derivation** (4 way Handshake)	3			
	Processing delay/Idle time	37			
5c	GTK Distribution**	0			
	Processing delay/Idle time	126			
6-	**DHCP**		31	36	389
	Processing delay/Idle time		4	5	6
7-	**Gratuitous ARP**		0	0	
	Total Time		***645***	**2360**	***4851***

Table 3.2: *Case 1* -Handover Phases of Netgear WG511T 108mbps wireless PC card

NB: From the *continuous unacknowledgment phase* until the *association phase* we used values obtained from using Azimuth Systems emulation environment.

Sub group presents the breakdown of *Higher Layer Authentication Phase*

* Average intentionally not taken see section 3.3.1.4.1 for details
** Average values

3.4.1 Analysis

3.4.1.1 Continuous Unacknowledgment Phase

The Netgear STA showed a very network friendly behaviour during this phase. We think that an average delay of 387 ms is comparatively lower. Based on its *handoff threshold criterion* the Netgear STA quickly detected the unavailability of the origin AP and was quickly able start looking for another suitable AP. This aggressive behaviour helps in minimizes the overall handoff latency, as otherwise this phase alone could be the many times longer then all the rest of the phases. We believed that Netgear showed hotspot/network friendly behaviour in this particular phase.

3.4.1.2 Scanning Phase

Although Netgear exhibited very aggressive behaviour in the previous phase but its behaviour in scanning phase was strangely different, average scanning delay of 1112 ms comparative to other STAs ,it was quite high. We noticed that Netgear was very reluctant in handing over to new AP, even after finding a new AP at good signal strength. We observed that Netgear performed multiple scanning rounds before finally proceeding with authentication and association.

Netgear scanned all the channels 4-5 times (on average) and on most occasions it received probe response from the *target AP* (to which it eventually connected) with very good signal strength in every round. We also observed that signal strength received from the target AP in the 4th or 5th round was generally lower than or equal to that in first round. This shows that a weak signal strength of the target AP wasn't the reason for the prolonged scanning phase. We observed this in our both testbeds. We think that with respect to this behaviour Netgear should be placed into family of *Home user friendly* STA's.

3.4.1.2.1 Processing delay / Idle time

This time is taken by the algorithm to select the best available AP. We have seen a *processing delay / Idle time* of 0 ms (i.e. a few micro seconds) or 1 ms on average , but we also noticed rarely exceptionally high *Processing delay / Idle Time* of up to 65 ms. We suspect that this delay should be attributed to the time required by the STA to calculate best channel (although it appears to sometime be a very large time). Thus suggests that there is other processing which has a higher priority once in a while, for example processing SNMP queries.

3.4.1.3 MAC Layer Authentication

Association time of 38 ms on average was observed. However lower values of 10-12 ms very frequently observed.

3.4.1.3.1 Processing delay / Idle time

Authentication is followed by a very short *Processing delay / Idle time* of approximately 1 ms which is quite understandable since at this stage the STA has to proceed to the *authentication phase* without any need for significant calculation.

Both authentication and association phases are processed by a UMAC and LMAC which are part of single module called SoftMac, this results in quicker processing from AP side. Lower MAC (LMAC) is embedded firmware to control the hardware, while the Upper

MAC (UMAC) provides additional functionality required for 802.11 devices. The various modules of AP are shown in figure 3.9.

3.4.1.4 Association Phase

On the average Association phase took comparatively little time, i.e., less than 28 ms, since only few messages are exchanged between the STA and the AP.

3.4.1.4.1 Processing delay / Idle time

Association phase is followed by a very long *processing delay / Idle time* which was nearly constant, the gap between the association response and the start of PMK derivation phase was 158ms and we observed that over many experiments this was almost constant, even by changing to different vendors model STAs, this remained nearly constant. Although we have very rarely seen lower values of upto 116 ms. We attributed this delay to the AP, as the AP was responsible for sending the EAP-Request identity to STA. However, in theory we expect that the STA could send an optional EAPoL-Start frame to initiate the 802.1X negotiation, but STA was never observed sending a EAPoL-Start frame. Unfortunately we did not had any means to force this EAPoL-Start packet to be sent by the STA, we suspected that probably usage of EAPoL start message would have shortened this delay .

We further investigated that how the AP processes various phases internally. In this regards a simplified diagram showing the AP's internal processes, modules, logical devices, transports, and interfaces is shown below.

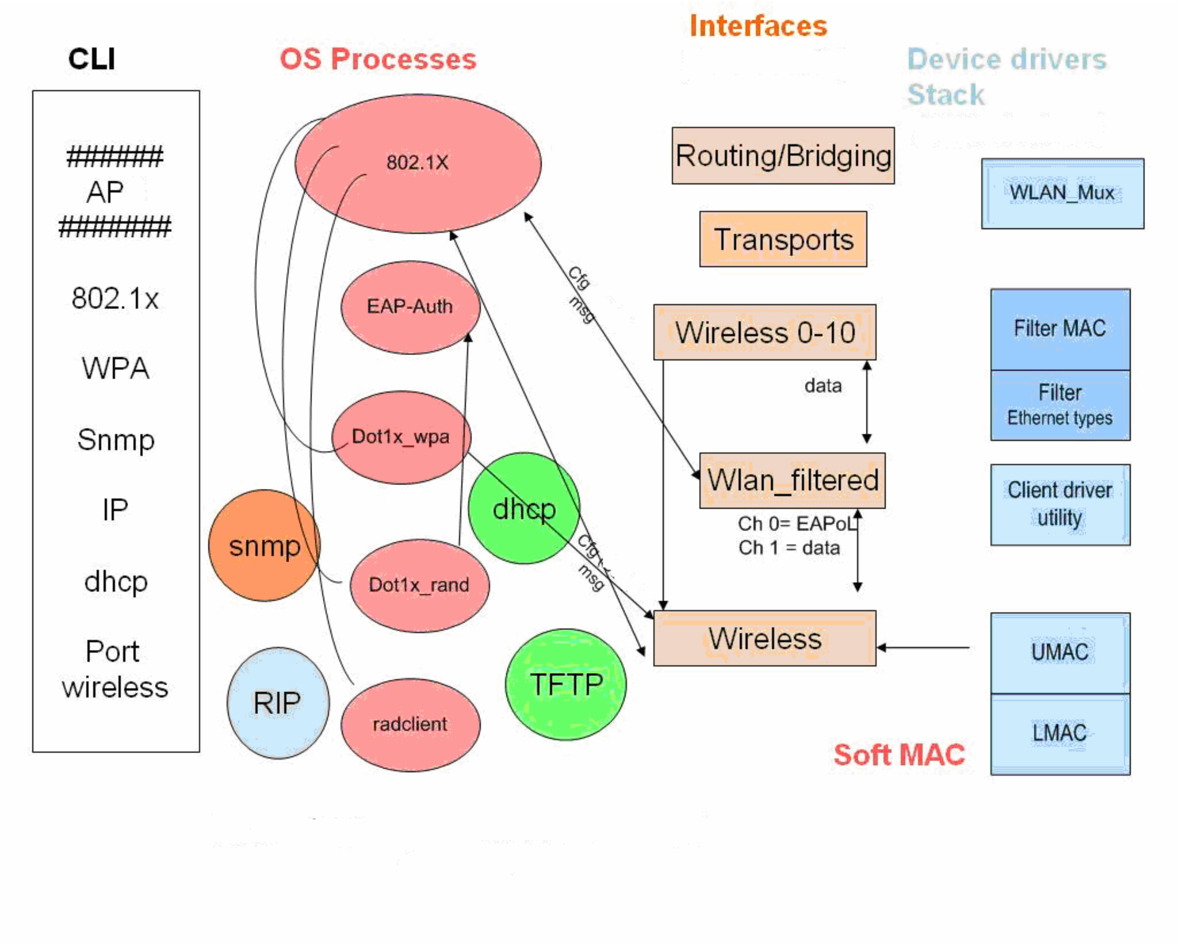


Figure 3.9 : Our Access Point internal software architecture (simplified) [37]

SNMP : Simple Network Management Protocol
DHCP : Dynamic Host Configuration protocol
TFTP : Trivial File Transfer Protocol
RIP : Routing Information Protocol

In simplified form our AP's software architecture could be divided into four layers where each layer is performing a specific task.

1. Device driver stack
2. Interfaces
3. OS processes
4. Command Line Interface

In the device drivers stack upper and lower MAC (UMAC and LMAC) constitute one single module called soft MAC [31]. Crossbow is a client utility or driver for the device [37].

Next to device drive stack is *Interfaces layer* which is responsible for passing filtered and non filtered data to OS processes layer. Interface layer is used for setting up the *routing* or *bridging* mode of the Access Point, setting up multiple SSIDs, and multiple transports by making use of logically expanded wireless ports (wireless 0-10). An interface layer utilizes separate propriety protocols for exchanging data and configuration messages with various application processes. Discussion of the Access Point's software architecture is beyond the scope of this thesis. However, more information could be obtained from the vendor, if required [37].

OS processes layer is responsible for *hosting* various modules (*applications* and *protocols)*, which are used to perform specific tasks. Each application or module may comprise of sub modules depending of complexity of task, they can, directly or indirectly, communicate with other layers. Examples of some of the modules/sub modules in OS processes layer are 802.1X, EAP-auth, SNMP module, DHCP module, RIP module radclient etc.

Finally, the fourth layer offers a command line interface to the users enabling them to communicate with the lower layers. Via CLI one can talk *indirectly* or *directly* to the firmware. Details of this Access Point's internal software architecture are outside the scope of this thesis.

The reason for comparatively low processing delays observed before *MAC layer authentication* and *association phases* is that these processes are both handled by a single module called "soft MAC" , i.e., with in devices drivers stack. A possible reason for the large delay between the *association phase* and the start of 802.1X (*PMK derivation phase*) is the fact that identity request message had to be initiated by a 802.1X module; higher up in the OS *processes layer* which passes through the wireless and wireless_ filtered port. We suspect that more time is spent either by the 802.1 X modules (and its sub-modules) or due to communication via *configuration interfaces* between wireless /wireless filtered and 802.1X. However, to verify this requires further debugging of each module, sub module and configuration interfaces involved, which were judged to lie outside the scope of the thesis.

3.4.1.5 PMK Distribution Phase

The pairwise master key distribution phase is the one of the longest phases of the whole process. This phase would be even worse if the certificate is larger. On Ethernet side, the normal MTU size is generally set to 1500 bytes- which poses a limit on certificate sizes without fragmentation. In our case server size certificate needed to be was 4710 bytes, and the STA certificate was 2863 bytes in size, so both of these messages were needed to be

fragmented (in to 4 and 3 fragments respectively). This phases added significantly to the overall handoff latency contributing on average 413.8 ms.
On further investigation we have noticed that during the PMK derivation total time taken on RADIUS side for this phase was significantly less then the time taken on WiFi side. There was generally a difference of 10-20 ms and the reasons for this are:

a) The initial identity request message is sent by the AP, rather then RADIUS server, and the response from the STA also reaches RADIUS after a short delay

b) The Last packet RADIUS Accept, takes sometime to reach the STA on WiFi side, then the STA sends a CTRL/ACK pair of frames which results in some extra delay on WiFi side.

In case of *certificates below the fragmentation size,* only 8 messages should have been exchanged between the STA and the RADIUS server, however, as noted earlier in our case 20 messages were exchanged.

We have observed that while repeating experiments occasionally a previously cached session ID, was used which helped to reduce the PMK derivation phase to **81 ms** (on the WiFi side), on further investigation we noticed that following this the STA and RADIUS server did not exchange the certificates and jumped directly to the change cipher suite step. In this case the time taken on RADIUS server side was lower then the WiFi side, due to reasons described earlier.

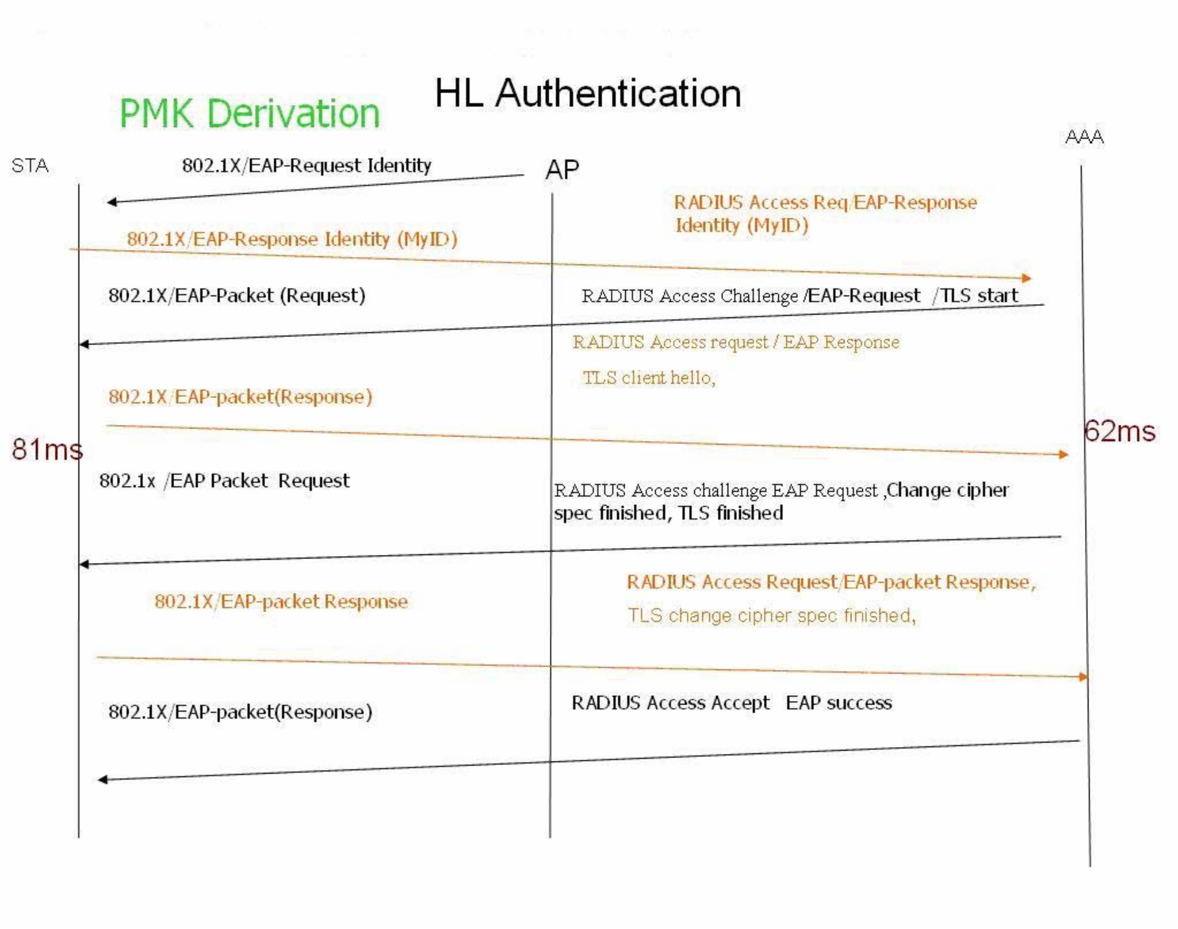

HL Authentication
PMK Derivation
STA
AP
AAA
802.1X/EAP-Request Identity
802.1X/EAP-Response Identity (MyID)
RADIUS Access Req EAP-Response Identity (MyID)
802.1X/EAP-Packet (Request)
RADIUS Access Challenge /EAP-Request /TLS start
RADIUS Access request / EAP Response
TLS client hello,
802.1X EAP-packet(Response)
81ms
62ms
802.1x /EAP Packet Request
RADIUS Access challenge EAP Request ,Change cipher spec finished, TLS finished
802.1X/EAP-packet Response
RADIUS Access Request/EAP-packet Response,
TLS change cipher spec finished,
802.1X/EAP-packet(Response)
RADIUS Access Accept EAP success

Fig 3.10 : PMK derivation without certificate exchange

3.4.1.5.1 Processing delay / Idle time

This step was also followed by a nearly constant *processing delay / Idle time* of 16 ms. It was also interesting to observe that change of STA to another vendor model did not effected the delay. We attributed this delay to AP side for processing required in sending next frame *EAPoL key* for start of Pairwise Transient key (PTK) derivation phase and in transfer of PMK from RADIUS server to AP.

3.4.1.6 PTK Derivation phase (4-way handshake)

In this phase 4 EAPoL key messages are exchanged to derive this took a nearly constant time (3 ms in most of the cases).

3.4.1.6.1 Processing delay / Idle time

Than the 4 way handshake is followed by another nearly constant gap of 37 ms which is on higher side as compared to Processing delay/Idle time followed by PMK derivation phase. As with earlier 2 nearly constant *Processing Delays / Idle Time periods* ,change of the vendor and model of the STA did not effected this delay , thus we believe again that this time was taken by AP to prepare for sending the next message (distribution of Group Transient Key (GTK) was due on AP). This time could be attributed to either to install temporal keys or to idle time. The *sub module* taking care of PTK derivation should further investigated. It is also worth mentioning that we often observed an encrypted broadcast message being sent out from STA→ AP during this period.

3.4.1.7 GTK Distribution Phase

GTK distribution was done very quickly and it took less then 1 ms on average.

3.4.1.7.1 Processing delay / Idle time

We observed three different behaviours for *post higher layer authentication* as noted section 3.2.6. The possible explanations for different behaviours are :

Case 1: usually 126-210 ms *delay* was observed prior to the start of the DHCP phase. As in **Intra ESS** handovers **DHCP phase shouldn't actually occur**, we suspect that during the overall handover process if **WLAN STA resets**, i.e., *brings up* the interface again this might result in a DHCP phase. We suspect that the longer *continuous unacknowledgment phase*, *scanning phase* and *PMK derivation phase* were the possible reasons for the **interface to be reset**. However, its worth mentioning that the STA was assigned the same IP address which was used during association with previous AP.

Case 2- On average 4 ms of *delay* was observed prior to the appearance of an ICMP exchange phase.

We suspect that in this case WLAN STA did not observe a link failure (i.e., the link going) down and hence an DCHP request was not required. It is worth mentioning that during this phase there was no ongoing voice traffic, reasons for not always having continuous voice traffic throughout the handover process are described in chapter 4

Case 3- On average a delay of 1 ms was observed prior to the arrival of first voice packet. However, occasionally we also observed the **first** voice packet arriving between the frames of the GTK distribution phase.

3.4.2 Post Higher Layer authentication

During the post higher layer authentication three type of behaviours were observed as described earlier in section 3.2.6. Table 3.2 shows only the results of case 1 (for case 2 and case 3 please see tables 3.5 and 3.6)

3.4.2.1 DHCP Phase

Generally in the DHCP phase 2 messages are exchanged; i.e. a DHCP request and DHCP response with the address assignment, but DHCP discover, offer, request and assignment messages were also seen very rarely. This phase usually took 33-37 ms in case of 2 messages, where the STA was always assigned the same IP which it held in the association with previous AP. We are not sure why Ajeet [21] and J.O. Vatn did not notice this but we suspect that Ajeet separately measured higher layer authentication time and added it on top of time required for layer 2 handovers. Moreover, Ajeet used pre-authentication which clearly reduces the overall handoff latency [21]. J.O. Vatn **theoretically** studied layer 3 handovers and but performed tests for only layer 2 handovers [12]. We also noticed that in layer 2 handovers the *DHCP phase* never occured.

3.4.2.1.1 Processing delay / Idle time

The DHCP phase is followed by a delay of 4-6 ms on average due to *early appearance* of Gratuitous ARP message to update the MAC forwarding table of the Distribution Side.

3.4.2.2 Gratuitous ARP Phase

This broadcast message requires only microseconds itself to be transmitted and is used by the STA to update the MAC forwarding table of the Distribution Side.

3.4.3 Statistical Analysis of the Netgear WG511T WLAN STA

We next statistically analyse the complete handover process. Netgear has extraordinary long scanning phase which results in a total handover delay of over 2 seconds, but at the same time Netgear has a very short *continuous unacknowledgment phase* as compared to both the Cisco and D-Link cards. Statistical analysis reveals that amount of time spent in processing delay/Idle time for the Netgear card ranges from 213-495 ms -- which is very significant. At minimum, it is 38% of the total handover time with 32% of the time being spent by the AP. This points the need to improve the software/firmware of the AP, for example better resource management *might* help reduce the over all delay by 20-30% in the best case . Fragmented packet exchanges of large certificates at the server and client side also increased the Pairwise Master Key (PMK) derivation delays and ultimately added greatly to overall handover delay.

The only difference between Case 1 and *Case 2/Case 3* is the **Post Higher Layer Authentication** *processing delay / idle time* due to status of WLAN interface. Table 3.5 and 3.6 shows average delays for the *cases 2 and 3.*

3.4.3.1 Case 1 –Netgear WG511T Handover Delay *(Average)*

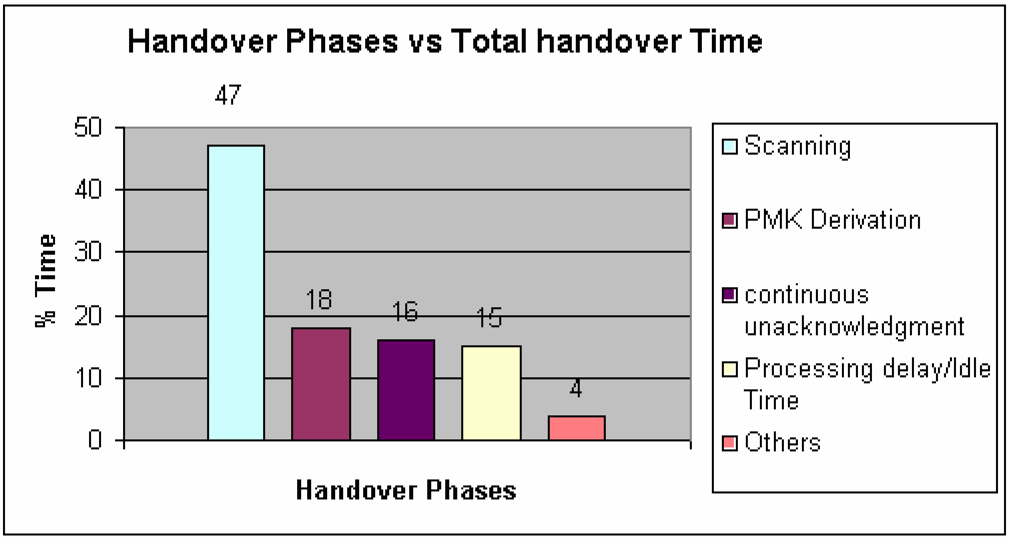


Figure 3.11 **:** Case 1 – Netgear WG511T Handover Delay *(Average)* Graph

Handover Phases	**Fraction of total handover**
PMK derivation	18 %
Processing delay/ Idle time	15 % (AP 9 %+ STA 6%)
Scanning	47 %
Continuous unacknowledgment	16 %
Remaining Others	4%

Table 3.3 : Case 1 – Netgear WG511T handover delays *(average)*

NB : The Netgear WG511T has an extraordinary long *scanning phase* and a very short *continuous unacknowledgment phase.*

3.4.3.2 Case 1 – Netgear WG511T Handover Delay *(Minimum)*

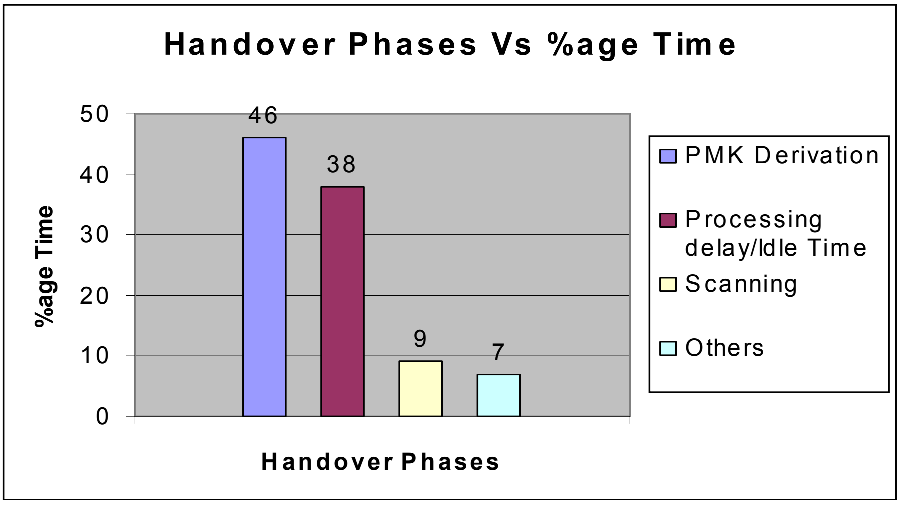


Fig 3.12 : Case 1 – Netgear WG511T Handover Delay *(Minimum)* by Graph

Handover Phases	**Fraction of total handover**
PMK derivation	46 %
Processing delay/ Idle time	38 % (AP 32 % + STA 6%)
Scanning	9 %
Remaining Others	7 %

Table 3.4: Case 1 – Netgear WG511T Handover Delay *(Minimum)*

NB : Proportion of time elapsed in *processing delay / idle time* is worth noting in this case.

3.4.3.3 Case 2 –Netgear WG511T Handover Delay *(Average)*

Step	**Phase**	**Sub group**	**Avg T (ms)**
1-	**Continuous unacknowledgment**		387
2-	**Scanning**		1112
	Processing delay/Idle time		*1*
3-	**MAC Authentication**		38
	Processing delay/Idle time		*1*
4-	**Association**		28
	Processing delay/Idle time		*158*
5-	**Higher Layer Secure Authentication**		416
	Processing delay/Idle time		*57*
5a-	PMK derivation (wifi)	413	
	Processing delay/Idle time	16	
5b-	PTK Derivation (4 way Handshake)	3	
	Processing delay/Idle time	37	
5c-	GTK Distribution	0	
	Processing delay/Idle time	4	
6-	**ICMP Exchange**		3
	Processing/wait		
	Total Time		**2201**

Table 3.5 : *Case 2* - Netgear WG511T average handover delays

NB : only difference from the case 1 is, Post Higher Layer Authentication delay is very short compared to OBSERVED *Processing delay/Idle time.*

3.4.3.4 Case 3 –Netgear WG511T Handover Delay *(Average).*

Step	**Phase**	**Sub group**	**Avg T (ms)**
1	**Continuous unacknowledgment**		387
2	**Scanning**		1112
	Processing delay/Idle time		*1*
3	**MAC Auth**		38
	Processing delay/Idle time		*1*
4	**Association**		28
	Processing delay/Idle time		*158*
5	**Higher Layer Secure Auth**		416
	Processing delay/Idle time		*57*
5a	PMK derivation (wifi)	413	
	Processing delay/Idle time	16	
5b	PTK Derivation (4 way Handshake)	3	
	Processing delay/Idle time	37	
5c	GTK Distribution	0	
	Processing delay/Idle time	1	
6	**First voice packet***		0
	Processing/wait		
	Total Time		**2198**

Table 3.6 : *Case 3* - Netgear WG 511T average delays with bidirectional voice traffic.

*Voice packet itself takes microseconds to be transmitted. But it's worth mentioning that occasionally we have seen first voice packet arriving **before** and **during** GTK distribution phase.

3.4.4 Statistical analysis Cisco Aironet WLAN STA handover delays

The Cisco aironet PC card had very different behaviour from the Netgear card with respect to *continuous unacknowledgment* and *scanning phases.* Cisco had an extraordinary long *continuous unacknowledgment phase,* almost 2-3 times longer then Netgear. On the other hand its scanning phase was quite normal where as STA scanned all the channels **only once**. The results obtained from the measurements of Cisco were similar to those of Ajeet Nankani [21] and J.O Vatn [12] where they both reported that Prism based STA's (including D-Link and Zyxel ZyAir) had a scanning time in the range of 210-250 ms. Ajeet also reported that before the start of the scanning phase approximately 70% packets of all packets were lost , this shows that it was *longest phase in* Ajeet's case also, where maximum packet loss occurred. However, Ajeet did not measure this delay as he didn't count this phase as part of over all handover process. Our measurements with the Cisco card further confirmed that 73% of the total handover time was spent in *continuous unacknowledgment phase* and hence this appears to be **most critical** (and lengthy phase) in overall handover process for this card. As duration of this phase is largely dependant on network coverage, its worth mentioning that we used **same configuartion** of Azimuth Systems, emulation enviroment (which offers interference free isolated chambers for AP's and both the STAs) for measurements of the Cisco and the Netgear delays; this clearly shows that Netgear has a more aggressive approach to handoff and it appears to have better criteria for calculating its handoff threshold. With respect to this criteria Cisco Aironet card doesn't exhibits *hotspot/network friendly* behavior but rather Cisco card exhibits more *network friendly* behavior then does the Netgear during the scanning phase.

During PMK derivation phase we noticed that the Cisco card sent the **EAPoL-start packet** to start this phase 6 ms after *association response* from the AP. Initially it appeared that 158 ms delay between *both the phases* has been eliminated with the usage of EAPoL-start packet. However, on further investigation we found out that the AP sent an *Identity request* **116 ms** after EAPoL-start message was received; which further confirmed that the processing delay was only on the AP side. We also did not notice the Cisco card sending any gratuitous ARP packet. Remaining phases were quite similar to those of the Netgear card, averge durations of various handover phases are presented below.

Step	**Phase**	**Sub group**	**Avg (ms)**	**Comments**
1	**Continuous Unacknowledgment**		3014	
2	**Scanning**		**261**	
	Processing delay/Idle time		*1*	
3	**MAC Auth**		**26**	
	Processing delay/Idle time		*1*	
4	**Association**		**28**	
	Processing delay/Idle time		*6*	
5	**Higher Layer Secure Authentication**		**537**	EAPoL start used
	Processing delay/Idle time		*248*	
5a	PMK derivation (WiFi)	505		(after 116 ms Identity request was sent by AP)
	Processing delay/Idle time	18		
5b	PTK Derivation (4 way Handshake)	13		
	Processing delay/Idle time	19		
5c	GTK Distribution	4		
	Processing delay/Idle time	211		
6	**DHCP**		**35**	
	Processing delay/Idle time			
	Total Time		**4130**	

Table 3.7: Cisco Aironet 802.11 a/b/g wireless adaptor average handover delays .

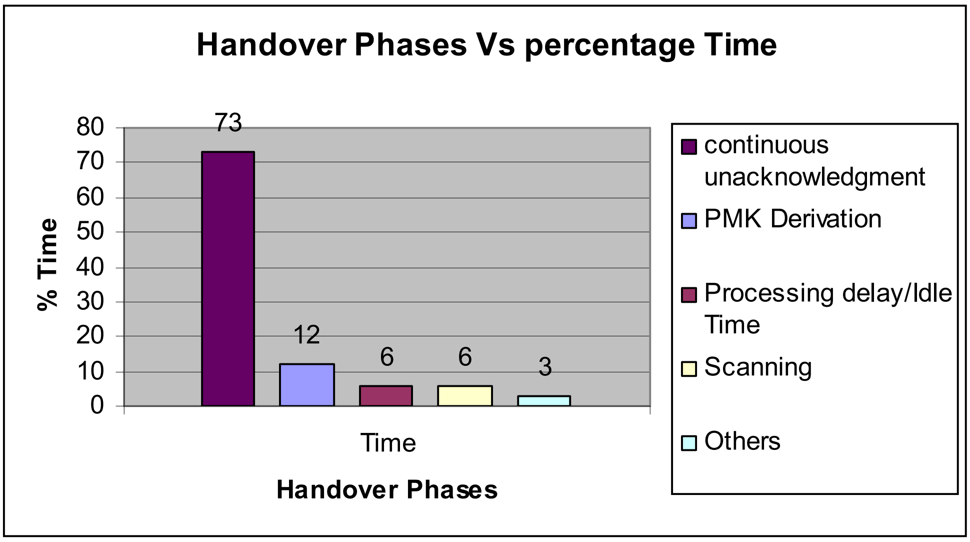


Figure 3.13 : Cisco Aironet 802.11 a/b/g wireless adaptor average handover delays

Handover Phases	**Fraction of total handover**
Continuous unacknowledgment	73 %
PMK derivation	12 %
Processing delay/ Idle time	6 %
Scanning	6 %
Remaining Others	3%

Table 3.8 : Cisco Aironet 802.11 a/b/g wireless adaptor average handover delays

3.4.5 Statistical analysis of a Linksys Handover delays *(average)*

The Linksys has a longest overall handover delay. It showed similar behaviour to the Netgear card during the scanning phase while its handoff threshold criteria was more or less the same as at used by the Cisco Aironet card (note that Linksys today is a division of Cisco). Remaining phases are similar to the Netgear card, except Post Higher Layer authentication *processing delay/ Idle time* which was on the average far greater in the Linksys card case than the Netgear card.

step	Phase	Sub group	Avg T (ms)
1	**Continuous Unacknowledgment**		2448
2	**Scanning**		**1364**
	Processing delay/Idle time		
3	**MAC layer Authentication**		**12**
	Processing delay/Idle time		*0*
4	**Association**		**8**
	Processing delay/Idle time		*158*
5	**Higher Layer Secure Authentication**		**448**
	Processing delay/Idle time		*493*
5	PMK derivation	441	
	Processing delay/Idle time	16	
5b	PTK Derivation (4 way Handshake)	3	
	Processing delay/Idle time	37	
5c	GTK Distribution	1	
	Processing delay/Idle time	438	
6	**DHCP**		35
	Processing delay/Idle time		4
7	**Gratuitous ARP**		**0**
	Processing delay/Idle time		1
	Total Time		**4971**

Table 3.9: Linksys dual band wireless A+G Notebook adapter Handover delays *(average)*

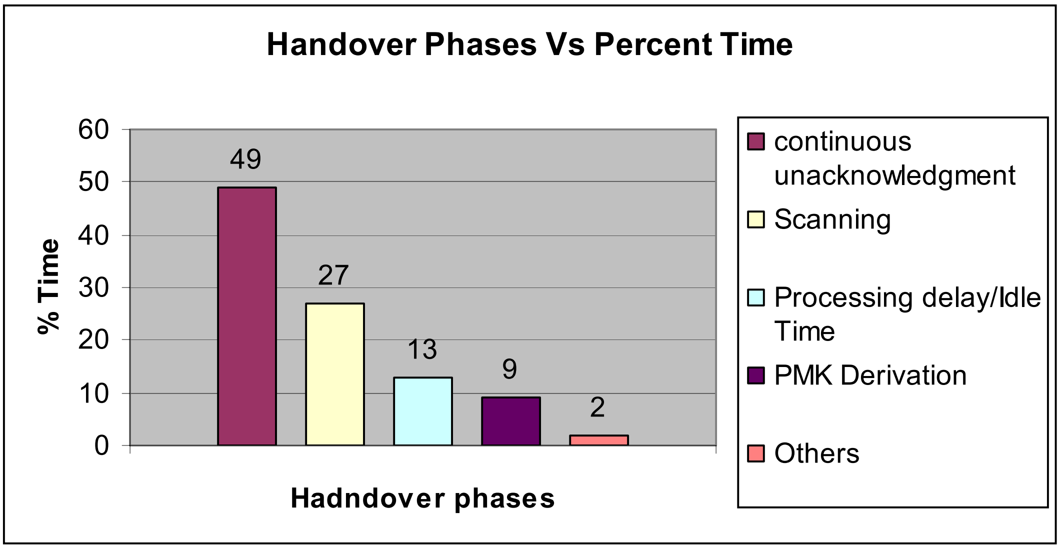


Figure 3.14: Linksys dual band wireless A+G Notebook adapter handover delays *(average)*

Handover Phases	**Fraction of total handover time**
Continuous unacknowledgment	49 %
Scanning phase	27 %
Processing delay/ Idle time	13 %
PMK derivation	9%
Remaining Others	2%

Table 3.10 :Linksys dual band wireless A+G Notebook adapter handover delays *(average)*

3.4.6 Suggestions yielding potential reduction of handover delays

3.4.6.1 Multimode STA's

We have seen that the handoff threshold criteria and scanning behavior of a particular STA varies widely, which in turn effects the VoWiFi roaming process. From a roaming perspective the Netgear has very good handoff threshold criteria, while Cisco exhibits significantly better performance in the scanning phase[2]. We believe that *best of both worlds* could be achieved by combining handoff threshold criteria of the Netgear card and scanning behavior of Cisco (or Lucent). Unfortunately having, *fixed criteria* for the both (*hotspots* and *home* users) target markets (and modes of operations) might not serve the requirement these markets. Therefore a *concept worth exploring* would be to use ***multimode STA's*** , where *one mode* will work well in *hotspot scenario* while other performs better in the *home user* mode, user (or their software agents) might opt for switching from *one mode to othe*r based on his/her present location or intended use or other criteria. By changing ***mode*** the *handoff threshold criteria* and *scanning behavior* of STA might be adjusted (in software/firmware) accordingly, Although there might be other implications involved in the implementation of this concept, it certainly offers an interesting option for 802.11 product vendors.

3.4.6.2 Network Facilitated Handovers.

major problem area for the handover process in 802.11 networks would be handoff threshold criteria; which is dependant on STA. Network controlled/facilitated handovers with the assistance of Access controllers might be a better choice for solving this issue although this will lead to higher *processing load* on Access Controller. 802.11k might be utilized in future to resolve this issue, where network controller can facilitate in providing/sharing necessary information with STA to help a seamless handover.

3.4.6.3 Pre-authentication

Pre-authentication might help reduce the PMK derivation delay to the minimum possible level, thus shortening the overall handover process. Pre-authentication should assist the access controller, as it alone cannot solve the *continuous unacknowledgment phase* problem.

3.4.6.4 Better resource management of AP's internal modules

Other then the time spent sending packets over the air, *processing delay/Idle time period* is also of significant importance, hence better resource management could reduce the handoff latency.

3.4.6.5 WLAN STA interface status

If status of WLAN STA interface **resets** during the lengthy handover process it might result in extra delay of several hundred milliseconds. Although it is difficult to tell exactly when STA interface may or may not go down, we suspect that reducing the overall delay might help maintain the WLAN STA interface in the UP state.

[2] we have seen J.O.Vatn [21] reporting that Lucent had even shorter scanning phase where average scanning time of STA was 81 ms

3.4.7 Limitations

- In 802.11 networks only the STA is responsible for *detecting roaming* we did not have access to the STA internals we were unable to be sure of some of our assumption for the causes of specific behaviour. Although we had access to the Access Point's internal activity but in order to understand handoff processing better we would need to have greater access to STA internals software/firmware to progress further.

- Our analysis was also limited to user of 3-4 STAs due to the project limited and the late availability of the roaming feature in the Azimuth System emulation environment.

- Roaming tests with WPA Enterprise were not possible in the Azimuth systems emulation environment (atleast in the automated mode) at our lab till date so we performed those tests in a semi-automated mode (which was quite time taking). However, we did open authentication tests in automated mode.

- concept of multimode STA is worth exploring. However commercial implications and the market impact need to be studied.

- In both methods of making measurments we only had two WLAN sniffers for listening to channels. We are not sure how the STA scanned rest of other channels during the scanning phase.

- The CommView sniffer missed frames in the first method of measurement i.e. real system testbed; which made it difficult for us to calculate the duration of the continuous unacknowledgment phase. However, although we measured the duration of *continuous unacknowledgment phase* in first method also, but in the end we used results obtained from the Azimuth Systems emulation environment.

- Due to the limitations of the Ericsson Access Point's current firmware release we couldn't study WPA2; which should be explored in future measurements, analysis of other EAP methods could be useful in order to better understand the handover process.

4 Voice Client and Services

4.1 The start of the IP –Telephony Era

IP telephony is changing the face of telecommunications in the modern world. Enterprises have adopted IP PBXs and IP telephones at such a rate that these equipment categories have become part of the telecommunications mainstream. Service providers are embracing VoIP equipment, such as soft switches and media gateway, as key elements within an IP multimedia subsystem (IMS) architecture that is currently being rolled out for to next-generation cellular networks. The market for enterprise IP Telephony and convergence is constantly reshaping.

A VoIP network carries voice traffic more efficiently switched than a switched circuit telephone network because IP telephony networks make better use of available bandwidth. In a public switched telephone network, for example, a dedicated 64 kilobits per second (kbps) end-to-end circuit is allocated for each call. In a VoIP network, digitized voice data is highly compressed and only needs bandwidth where there is voice activity and carried in packets over IP networks. Using the same bandwidth, a VoIP network can carry many times the number of voice calls as a switched circuit network and provide better voice quality. The savings realized in using VoIP networks are often passed onto users in the form of lower costs. [31]

The enterprise voice, IP Telephony and convergence market is undergoing unprecedented change, and customer needs and behaviours are changing along with it. Despite the wide range of value added services that IP telephony can offer today its ability to offer low cost alternative to circuit switch telephony that directly effects the *user experiences* and *expectations* for IP telephony service. Patience and tolerance (towards packet loss) of an IP telephony user today is **better** than for traditional circuit switch customer due to much lower costs involved. Customers might be ready to compromise a bit at atleast for certain calls , in return for significantly lower costs.

4.1.1 Threats for seamless handovers in VoWiFi handovers:

Handover duration within an IEEE 802.11 LAN can take a long time, especially if we are using EAP-TLS under WPA. Two major types of threats to voice traffic during such handovers are **Packet loss** and **call tear down**.

Packet loss occurs as fraction of time elapsed in the different phases the tolerance of the call to these losses is a function due to the voice CODEC used during a handover in WPA based scenarios. Our experiments examined these delays in the context of Statistical analysis of packet loss was studied in great detail by J.O. Vatn [12] and Ajeet Nanakani [21] but we have not seen much (if any) prior work on *call tear down* until this thesis. Call tear down phenomena will be described in following sections.

We think that usage of *traffic generators emulating voice like traffic* might be a good idea for exploring packet loss during handover, even though it *ignores* the impact of call tear down during the handover process. In our experiments using our testbed we have tried to use various voice services to analyse this impact. It was interesting to notice that many voice sessions /services they do not survive the long handovers delays, and thus voice session might be terminated before the completion of the handover, thus we think that results based on traffic generators might be overestimated.

4.2 Tests with Voice clients

We used a number of different voice clients to observe the impact of handover on voice traffic. The specific PC based software voice clients and services used were:

- Microsoft MSN messenger version 7.0 built 7.0.0813 with MSN voice service (SIP), (see section 4.2.1)
- Skype version 1.4.0.78 (see section 4.2.2)
- X-lite version 2.0 release 1103 m build stamp 14262 with IPTEL account (see section 4.2.3)
- X-lite version 2.0 release 1103 m build stamp 14262 with an IMS account at Ericsson IP Multimedia Subsystem Telephony (see section 4.2.4)

Our study was not intended to understand the behaviour of each of these commercial clients in detail, but rather we wanted to measure the impact on handover of the underline protocols used. In this regard we have benefited from prior studies and work done examining commercial protocol behaviour, together with our own observations. We believe that this also provides the basis for additional **future work**.

Testbed used for these experiments is the same as described in section 3.1.1. *Sniffer 2,* on fixed network, was used as the *caller* while the mobile STA was used as the *callee* in most of the test cases. We used the term **Roaming Friendly** to differentiate the behaviour of various voice clients. We rated a client as **Roaming Friendly** when it never tears down call during the handover and we used the term **Non Roaming friendly** for those clients who were not able to maintain the an ongoing voice session during a handover and finally at some point during this process, terminated the call. We discovered that very nature of underline *protocols used* itself is of significant importance and should be taken into account when measuring the impact of handover on voice.

4.2.1 MSN Messenger

Microsoft's MSN messenger uses SIP signaling for establishing and tearing down a voice session and uses RTP over UDP for transporting media. To **log into** the MSN Passport network *most* of the communication takes place over TCP. To make an outgoing connection MSN uses port 1863 *(officially)*, although there are many places in the protocol where alternate ports could be specified, so this could change. MSN uses Microsoft Network Messenger (MSNM)'s protocol for communication between MSN client and a server and it uses Microsoft Network Client Protocol (MSNC) for communication between clients. In order to understand the impact of handovers on *call teardown* it is important to understand the *login / logout* procedure of MSN messenger. MSN uses **Notification Server, Switch Board, Dispatch Server,** and many other elements for this purpose. A brief overview of this process is given below.

The first step in an MSN Messenger session is logging into a **Notification Server.** The main purpose of the notification server is to handle presence information maintained about yourself and the principals whose presence you've subscribed to. If you have previously learned the IP address of a notification server, you can connect directly to that server. Otherwise, you must connect to the "dispatch server". The official client uses `messenger.hotmail.com`, port 1863 to connect a dispatch server (for direct and SOCKS-based connections) and `gateway.messenger.hotmail.com` port 80 as the dispatch server for HTTP connections.

To log off of the Notification Server the easiest way is to simply close the TCP socket. The *proper* way to log off is by sending the `OUT` command to the Notification Server with no parameter and no *Transaction ID (TrID).* The server will close the connection soon.

The connection to a notification server is the basis of MSN Messenger session, as it handles your presence information, i.e., if you are disconnected from the notification server, you are no longer online to your buddies. The notification server also performs some other services notifying you about new e-mail in your hotmail inbox and letting you create new or join existing switchboard sessions. A switch board session is used for chatting between peers or for file transfer etc. When you're directed to join a switchboard session, you should open a new connection to the switchboard, and while maintaining a connection to notification server.[33]

User must login to the MSN Passport Network in order to use, any of the value added services, including voice. The message flow diagram is shown below.

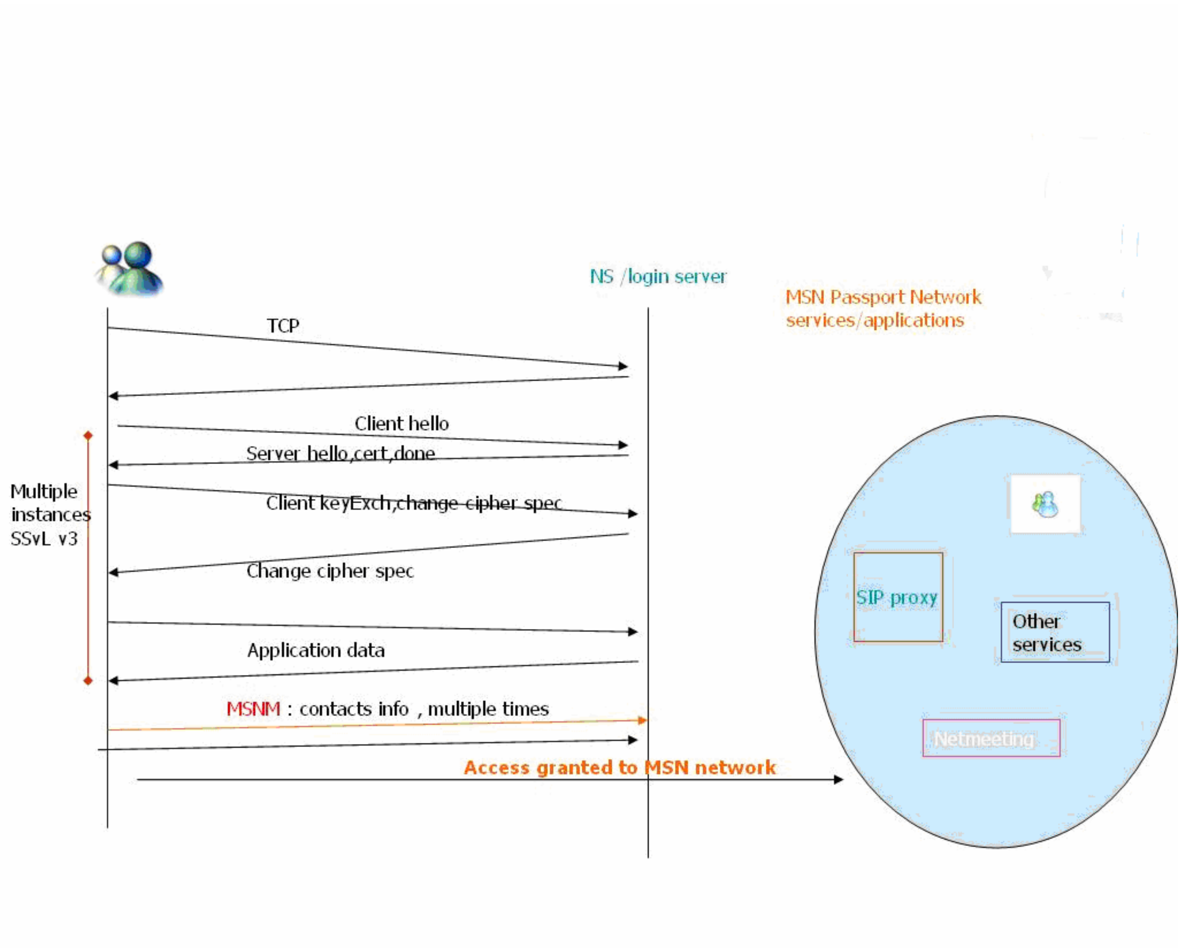


Figure 4.1 : Simplified MSN login procedure [3]

[3] MSN logo is property of Microsoft Network

For both voice and video sessions the client uses SIP and SDP. This SIP user agent is built-in MSN messenger. To create an audio session client sends SIP_A and for video SIP_V [33].

Once the voice session is established RTP data streams can flow from both sides. We initiated roaming from the *Origin AP* to the *Target AP* after establishing the voice session. We noticed that voice session was not able to survive the handover. It is because MSN client **monitors** amount of packet loss (with the help of RTCP reports) and after a *certain threshold* the roaming MSN Client sends a **RTCP "GOOD BYE"** message which is quickly followed by a **SIP BYE,** later we see a finally, **MSNM OUT.** This behavior is fundamentally different then the X-Lite as X-Lite doesn't take into account amount of packet loss as a criteria for terminating the an ongoing voice session.

Although we think that major problem for the MSN was the length of the *continuous unacknowledgment phase* and the *PMK derivation phases*. We observed that most of the time the user agent logged out **during** or **before** the PMK derivation phase. Unfortunately we couldn't measure the total duration before it logged out due to the fact the CommView (WiFi sniffer) occasionally lost packets which made it difficult to measure the duration of the *continuous unacknowledgment phase* and in turn the total time before MSN client finally logged out.

We rated this UA as a **Non Roaming friendly client.** This is another **primary reason** that one can observe ICMP Exchanges phases during handover or DHCP and Gratuitous ARP phases as there was *no voice traffic left* . when using traffic generators there is no threat to call termination, , as it is the client which is terminating the call for its own reasons, hence it behaves differently then in this case then when using active calls.

Calculation of the exact threshold for termination of the call used by MSN was outside the scope of this thesis. However, is an obvious measurement, for future work and also it would be interesting to see the results with WPA2 and Pre-authentication.

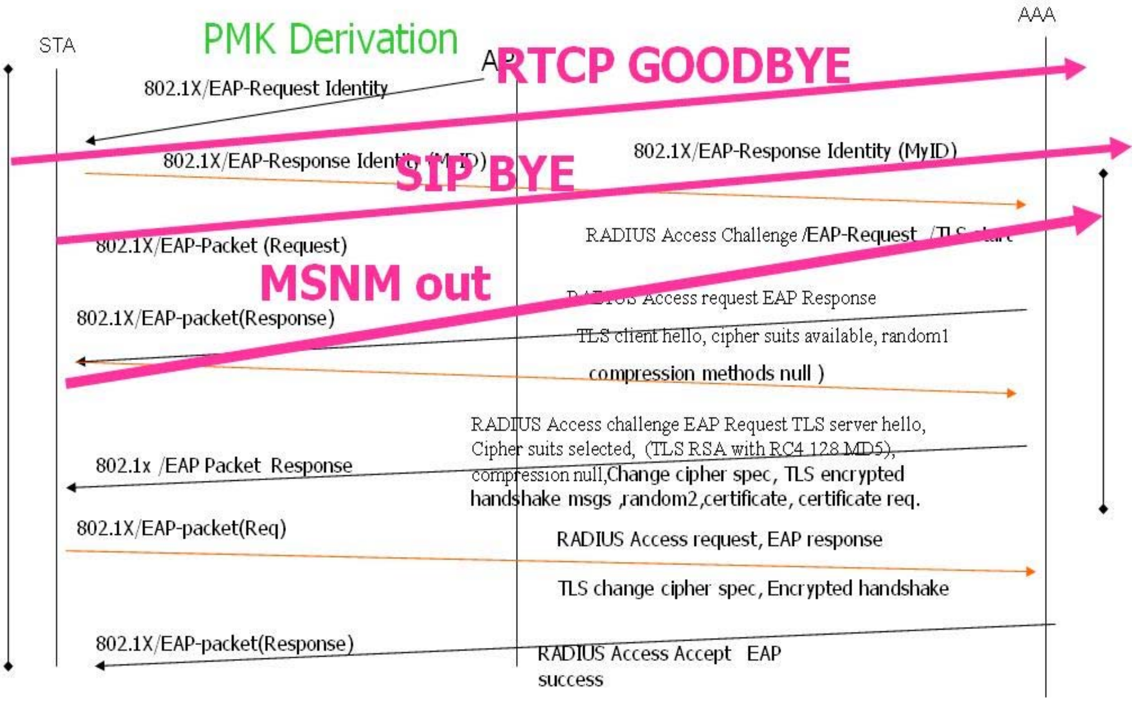

STA
PMK Derivation
AP
AAA
RTCP GOODBYE
802.1X/EAP-Request Identity
802.1X/EAP-Response Identity (MyID)
SIP BYE
802.1X/EAP-Response Identity (MyID)
802.1X/EAP-Packet (Request)
RADIUS Access Challenge /EAP-Request /TLS start
MSNM out
RADIUS Access request EAP Response
802.1X/EAP-packet(Response)
TLS client hello, cipher suits available, random1
compression methods null)
RADIUS Access challenge EAP Request TLS server hello,
Cipher suits selected, (TLS RSA with RC4 128 MD5),
compression null,Change cipher spec, TLS encrypted
handshake msgs ,random2,certificate, certificate req.
802.1x /EAP Packet Response
802.1X/EAP-packet(Req)
RADIUS Access request, EAP response
TLS change cipher spec, Encrypted handshake
802.1X/EAP-packet(Response)
RADIUS Access Accept EAP
success

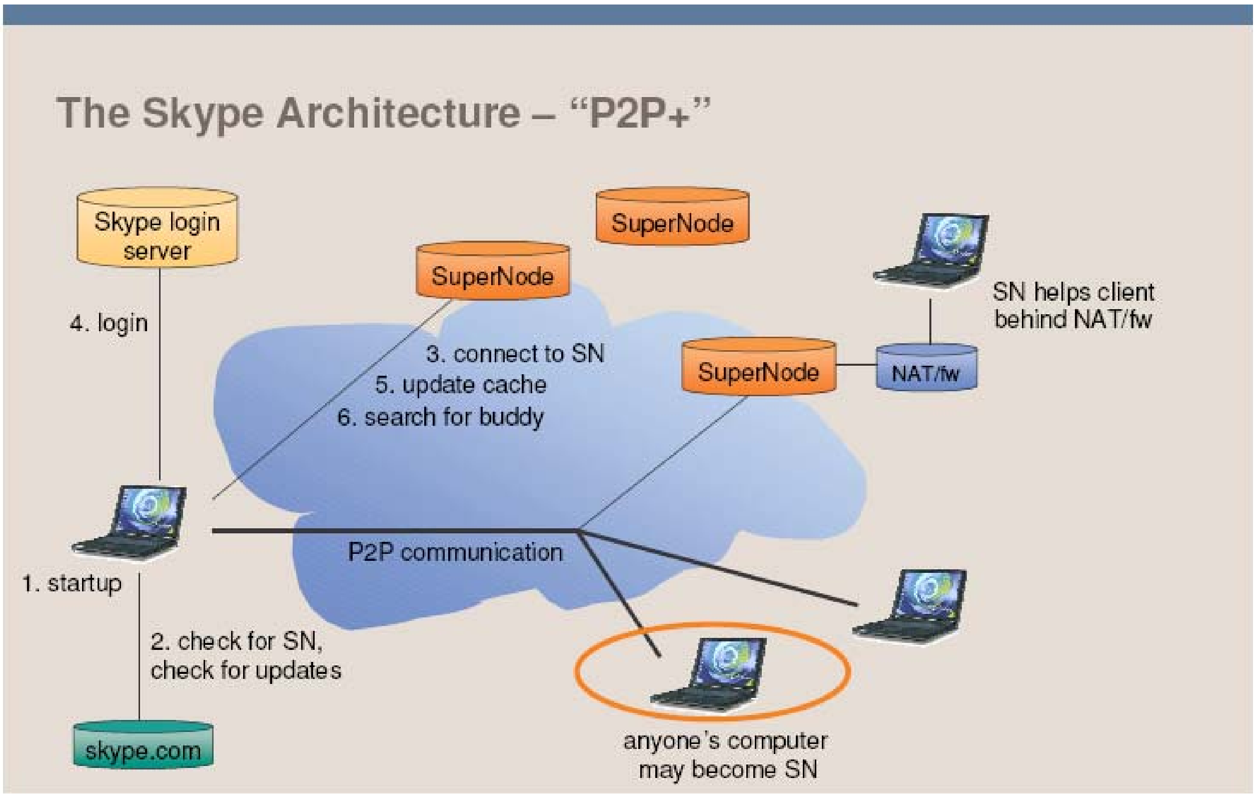
Figure 4.2 : MSN during handover terminating voice session.

4.2.2 Skype PC Client

Another client which was tested was Skype PC client. Skype is available for Microsoft Windows PC, Apple Macintosh MAC, Linux and Pocket PC. We have only tested Skype on Microsoft Windows and our analysis is based only on Skype PC client.

Although SKYPE is a propriety solution. However, there is some prior worl examining Skype by Salman A. Baset [34] and Carlos Marco Arranz [35. Salman A. Baset in his paper -“An Analysis of the Skype Peer-to-Peer Internet Telephony protocol explains that “ Skype is a P2P IP telephony solution utilizing two types of nodes in an overlay network : ordinary hosts and super nodes (SN). An ordinary host is a **Skype client** that can be used to place voice calls and send text messages. A **super node** is an ordinary host’s end-point in the Skype network. Any node with a public IP address having sufficient CPU, memory, and network bandwidth is a candidate to become a super node. An ordinary host must connect to a super node and must register itself with the **Skype login server** for a successful login. Although not a Skype node itself, the Skype login server is an important entity in the Skype network as user names and passwords are stored in there and user authentication at login is also done at this server. This server also facilitates NAT and firewall traversal.

Skype uses a wideband CODEC which allows it to maintain reasonable call quality given an available bandwidth of 32 kb/s or more. It uses TCP for signaling, and both UDP and TCP for transporting media traffic. Signaling and media traffic are sent on the same ports. The Skype architecture is shown in the figure below:



The Skype Architecture – "P2P+"

Figure 4.3 : Skype Architecture [36][4]

Skype uses its Global Index (GI) technology to search for a user. After locating a user the Skype client exchanges the following messages to establish a call (see figure 4.4)- detailed discussion of this process can be found in S Baset's report [34].

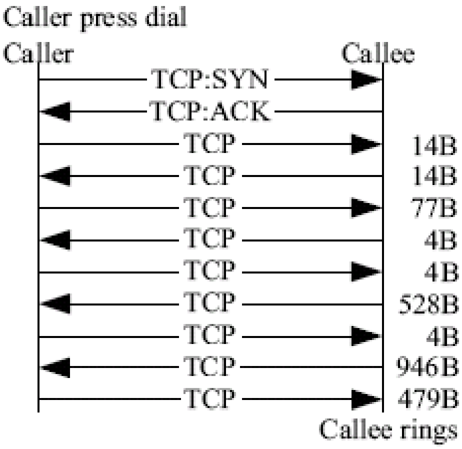


Figure 4.4 : Skype 's Call establishment procedure [34]

[4] With thanks to David Partain [36] for providing this image

After a callee picks up the phone the media starts flowing largely over TCP. At the end of session we observe a couple of TCP messages which are used to tear down the call.

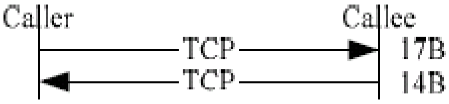


Figure 4.5: Skype's call tear down procedure [35][5]

Skype transports media over TCP and easily detects an unavailable network connection after **not** receiving number of TCP ACK within a bounded time. Moreover, it uses *keep alive* messages to detect the network availability. We think that Skype incorporates link layer information into the application when making a decision to teardown the call. We have noticed that during handover Skype PC client was tearing down the call in middle of the PMK derivation phase, hence doesnot not appeared to be a roaming friendly client. We found the Skype PC client to be a **Non Roaming Friendly client.**

We suspect that the longer *continuous unacknowledgment* and the *PMK derivation phases* are problem areas for Skype. Unfortunately the exact length of duration for this call teardown couldn't be measured due to the reasons given earlier in the section 3.4.1.1.

However its worth mentioning that Skype is both *propriety* and uses an *encrypted protocol*, thus it is very difficult to understand its behavior and to understand its underlying mechanism. As Skype *moves into VoWiFi business* they certainly should consider these factors before launching a Vonage like service using hardware based VoWiFi Skype phones.

Note : This report **only used Skype PC client** for testing, thus the results might not apply to other variants of Skype.

[5] Thanks to Salman Baset for providing figure 4.5 and 4.6

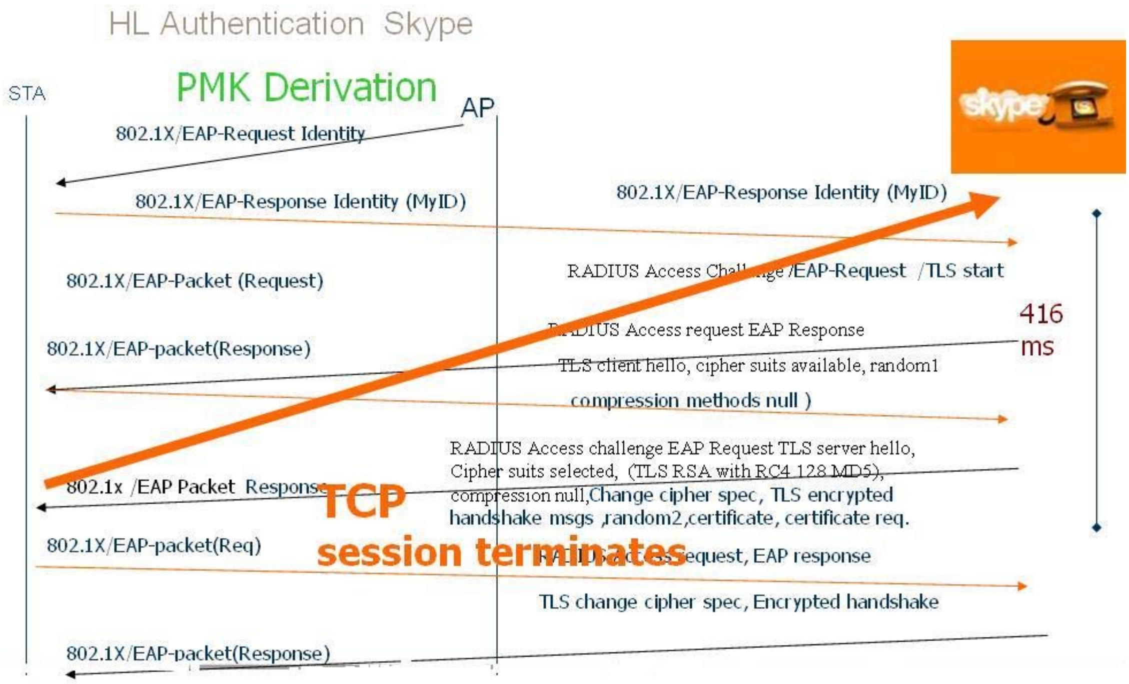

HL Authentication Skype
PMK Derivation
STA
AP
skype
802.1X/EAP-Request Identity
802.1X/EAP-Response Identity (MyID)
802.1X/EAP-Response Identity (MyID)
802.1X/EAP-Packet (Request)
RADIUS Access Challenge /EAP-Request /TLS start
416
ms
RADIUS Access request EAP Response
802.1X/EAP-packet(Response)
TLS client hello, cipher suits available, random1
compression methods null)
RADIUS Access challenge EAP Request TLS server hello,
Cipher suits selected, (TLS RSA with RC4 128 MD5),
compression null,Change cipher spec, TLS encrypted
handshake msgs ,random2,certificate, certificate req.
802.1x /EAP Packet Response
TCP
session terminates
802.1X/EAP-packet(Req)
RADIUS Access request, EAP response
TLS change cipher spec, Encrypted handshake
802.1X/EAP-packet(Response)

Figure 4.6 : Skype's call termination during PMK derivation[6]

[6] Skype logo is property of M/S Skype

4.2.3 X-Lite and an IPTEL account

In these tests we used Xten Network, X-lite version 2.0 release 1103 m build stamp 14262 and a standard account from IPTEL www.iptel.org .The IPTEL's **iptel.org** is a VoIP portal promoting VoIP technologies and was created by Germany's national research institute "Fraunhofer Fokus". The **iptel.org** also distributes VoIP software developed by Fraunhofer Fokus. Their SIP Express Router (SER), offers a basic SIP-based VoIP platform.

X-Lite is a product of Xten Networks which recently was bought by CounterPath Solutions, Inc. Xlite is a very popular free voice client ; i.e. it is a SIP user agent which offers a range of functionalities and features for communication with SIP based platforms. A simplified message flow diagram of logging into IPTEL SER (Sip Express Router) by X-Lite and making a call using has been shown below.

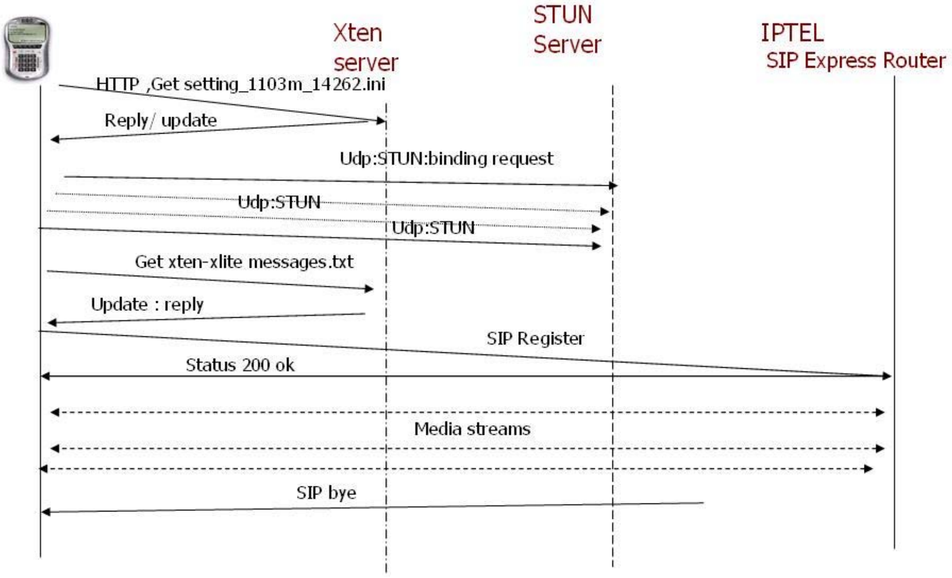
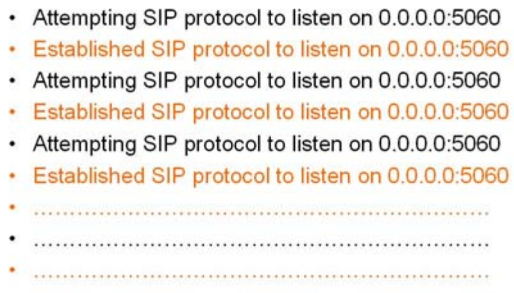


Figure 4.6 : Registering and making a call usingX-lite and an IPTEL account. [7]

The simplified message flow diagram illustrates the login procedure of an X-Lite to the IPTEL SER server. After discovering the Xten server the X-Lite user agent client (UAC) asks for a software update from the Xten server (in our case it never succeeded in finding the required update from this server).Next the UAC requests a STUN binding from the STUN server and finally registers with IPTEL's SER. Following this RTP streams are used during a call to send continuous media data. During a handover we noticed the UA doesn't

[7] X-Lite logo is property of M/S CounterPath Solutions

utilizes packet loss as indicator to send a SIP BYE and hence **never** terminates the call. during a period when there is no connectivity it keeps on attempting to connect *to a default network address on port 5060* (0.0.0.0:5060) and keeps on establishing a connection on 0.0.0.0:5060 as shown in the figure below.



```
• Attempting SIP protocol to listen on 0.0.0.0:5060
• Established SIP protocol to listen on 0.0.0.0:5060
• Attempting SIP protocol to listen on 0.0.0.0:5060
• Established SIP protocol to listen on 0.0.0.0:5060
• Attempting SIP protocol to listen on 0.0.0.0:5060
• Established SIP protocol to listen on 0.0.0.0:5060
• ...........................................................
• ...........................................................
• ...........................................................
```

Figure 4.7 : X-Lite during *no connectivity* period

This user agent sends a SIP BYE message upon explicit hang up by either the *caller* or the *callee,* hence it can survive even very long handover delay. In our tests of handover under WPA using EAP-TLS it survived **all** the tests. Our observation of this UA showed that it pass through all the phases **without** tearing down the call. It is important to point out that packet loss may always occur and counter measures such as *pre-authentication*, etc. should be used to minimize that effect of such packet loss. So with respect to call termination criteria X-Lite was the Most **Roaming Friendly** Client.

4.2.4 X-Lite and IMS Account at Ericsson IMT

Using X-Lite via an IMS proxy showed **Roaming Friendly behavior** similar to the behavior of X-Lite and IPTel *(i.e. never unnecessarily tearing down the call*) and hence it was rated as the **Most Roaming Friendly** solution.

4.3 Analysis

Various voice clients and services exhibit different behaviors with respect to their call tear down procedure. This behavior is related to underlying protocol used and hence some solution proven to be **More Roaming Friendly** than others**.** However, we believe that evaluation are unlikely to be static as more work continues to support *VoWiFi roaming.* Our work in evaluating roaming friendly behavior should be seen as **seed work** due to time limitations we couldn't look into further details. It would be interesting to compare other voice services and agents. The main threat to call tear down is probably the *continuous unacknowledgment phas*e which is far longer than the *PMK derivation phase*.

5 Conclusions and Future work

5.1 Conclusions

In this thesis we have examined the intra ESS handover process of 802.11networks in detail with emphasis on WPA Enterprise specific scenario. We concluded that STA's could be divided into atleast two categories with respect to their *handoff threshold criteria* and *scanning behavior* -- which are set by tuning various parameters in STA's software. Both of these **modes** have their own target markets and commercial implications. These modes were :

a) Single AP / Home user friendly mode
b) Network / Hotspot friendly mode

We believe that in order to offer a single optimized solution for both *Home* and *Hotspot* markets, the concept of **multimode STA** is worth exploring. The user may be switching the *mode* of STA from application (or it might be done automatically for them) ; which in turn might effect in changing *criteria* within the STA's process or firmware layer accordingly.

We analyzed all the handover phases in **detail** from both the Access Point as well as STA's perspectives. We noticed both the time spent sending packets *over the air* but also the time spent in *processing delay or remaining idle* were significant in VoWiFi handovers. We associated all such delays with either the AP or STA's internal activity.

The impact of fragmented packets on overall handoff latency is significant and has been studied in detail. We noticed that *during the Post higher layer authentication phase* the STA can exhibit different behavior depending on status of the STA's interface.

We concluded that usage of traffic generators for analyzing VoWiFi handover process might be overestimating as somee VoIP clients may **terminate** the call during the handover process depending on their underlying protocol. We rated several clients as Roaming and Non Roaming friendly with respect to their call teardown behavior. To our knowledge there was no prior work on this subject.

In the end we have given several suggestion (in section 3.5.3) to reduce overall handoff latency.

5.2 Future work

- We associated various handover phases with the Access Point's internal activity but unfortunately we did not have the access to the STA's internals. As in 802.11 networks the STA is completely responsible for *detecting the roaming*, it would be interesting to compare the *handoff threshold criteria* of various STA's based upon the actual code and parameters used. Study of handoff threshold algorithm in precise details could be very active area for future research.

- Our analysis was limited to 3-4 STA's due to time limitations and the late availability of the roaming feature in the Azimuth System emulation environment. A wide range of STA's should be tested with regard to *continuous unacknowledgment* and *scanning phase.*

- Concept of multimode STA's (for handoff decisions) is worth exploring, where a single STA can satisfy the requirements of both, the hotspot and the home user, markets. However industrial implications and market dynamics needs to be studied.

- One of the most important areas for future work could be studying call teardown procedure of various VoIP clients

- Due to the limitations of the Access Point's current firmware release we couldn't study WPA2 which should be explored once this possible. Analysis of other EAP methods could also be useful in order to better understand the handover process.

References

[1] ISO/IEC and IEEE Standard “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications” (as supplemented), 1999

[2] Voice over IP definition –at whatis.com http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci214148,00.html , 22 July 2005

[3] James W. Truelove, “Build Your Own Wireless LAN”, Berkeley, CA, USA: McGraw-Hill Osborne, 2002. p 174.

[4] J. Rosenberg and G. Camarillo , ‘SIP: Session Initiation Protocol’. RFC 3261 http://www.ietf.org/rfc/rfc3261.txt?number=3261 IETF June 4, 2005

[5] G. Q. Maguire Jr., ‘2G1325/2G5564 Practical Voice Over IP (VoIP): SIP and related protocols, Spring 2005, Period 3’, Lecture notes. http://www.imit.kth.se/courses/2G1325/VoIP-2005.pdf 29 June, 2005

[6] Khurram Jehangir Khan and Ming-Shuang Lang, ‘Voice over Wireless LAN and analysis of MiniSIP as an 802.11 Phone’, Final report for 2G1330 Wireless and Mobile Network Architectures Department of Microelectronics and Information technology, Royals Institute of Technology (KTH) Sweden. ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/ 2 July 2005

[7] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg , ‘SIP: Session Initiation Protocol’. RFC 2543 http://www.ietf.org/rfc/rfc2543.txt?number=2543 IETF , 22 July 2005

[8] M. Handley and V. Jacobson ‘SDP: Session Description Protocol’. RFC 2327 http://www.ietf.org/rfc/rfc2327.txt?number=2327 4 July , 2005

[9] John Vacca, “Wireless Data Demystified”. Blacklick, OH, USA: McGraw-Hill Professional, 2003. p 112. http://site.ebrary.com/lib/kth/Doc?id=10045612&ppg=136 2 July 2005

[10] Syngress Publishing(Editor). “Managing and Securing a Cisco Structured Wireless-Aware Network” Rockland, MA, USA: Syngress Publishing, 2003. p 429.

[11] IEEE Standard for Information technology--Telecommunications and information exchange between system--Local and metropolitan area networks Part 11: Specific requirements Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Medium Access Control (MAC) Security Enhancements, July 2004

[12] Jon - Olov Vatn , “IP telephony: mobility and Security” http://www.diva-portal.org/diva/getDocument?urnnbnsekthdiva-260-1fulltext.pdf (10 June 2005)

[13] WiFi Alliance http://www.wi-fi.org , 21 September 2005

[14] Juan Carlos Martín Severiano, "IEEE 802.11b MAC layer's influence on VoIP quality parameters", Master Thesis, Department of Microelectronics and Information technology Royals Institute of Technology (KTH) Sweden, ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/ 14 June 2005

[15] WPA and WPA2 Implementation White Paper "Deploying Wi-Fi Protected Access (WPA™) and WPA2™ in the Enterprise" , http://www.wi-fi.org/OpenSection/protected_access.asp 22 Oct 2005

[16] IEEE. Get IEEE 802 home Web page: http://standards.ieee.org/getieee802/ (12 July 2005).

[17] 802.11e standard, IEEE 802.11e ,Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), Oct 2005.

[18] The shmo group, Airsnot download http://airsnort.shmoo.com/ 13 Dec 2005

[19] Jouni Malinen, Linux hostap driver. http://hostap.epitest.fi. 9 July 2005

[20] Jon-Olov Vatn. "An experimental study of IEEE 802.11b handover performance and its effect on voice traffic." Technical Report TRITA-IMITTSLAB R 03:01, Telecommunication Systems Laboratory, Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH) Stockholm,Sweden, July 2003. http://www.imit.kth.se/˜vatn/research/handover-perf.pdf. 25 July 2005

[21] Ajeet Nankani,"Horizontal Handoffs within WLANs" ,Master Thesis, Department of Microelectronics and Information Technology, Royal Institute of Technology (KTH) Sweden ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/ 28 August 2005.

[22] Sangho Shin, A.S. Rawat, and Henning Schulzrinne. "Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs". In ACM MobiWac'04, oct 2004. Philadelphia, Pennsylvania, USA. http://portal.acm.org/ft_gateway.cfm?id=1023788&type=pdf 26 July 2005

[23] IEEE Std. 8021D, 1998 Edition, IEEE Standard for Information technology Telecommunications and information exchange between systems .Local and metropolitan area networks Common specifications Part 3: Media Access Control (MAC) Bridges, 1998.

[24] 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://www.it.kth.se/~hvelayos/publications.shtml 21 July 2005

[25] Airopeek, WLAN analyzer from WildPackets, http://www.airopeek.com
6 December 2005

[26] Azimuth Systems, users documentation www.azimuthsystems.com , 18 Nov 2005

[27] Jeff Thomas, *Network World online*, 23 June2003
http://www.networkworld.com/news/tech/2003/0623techupdate.html
20 June 2005.

[28] David John Oates, Founder and Developer of Reverse Speech Technologies
http://www.reversespeech.com/Simple_Examples.htm 11 September 2005

[29] B. Aboba, and D. Simon, Microsoft PPP EAP TLS Authentication Protocol
October 1999 http://www.faqs.org/rfcs/rfc2716.html 17 Oct 2005

[30] IEEE Std 802.11F, IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation, June 2003.

[31] The Prism54 project , Soft MAC Module from www.prism54.com , 11 Oct. 2005

[32] Performance technologies http://www.sctp.info/products/prodgroupsignaling.html
9 Nov 2005

[33] Mike Mintz, unofficial website for MSN, http://www.hypothetic.org/docs/msn/ ,
2 Oct 2005

[34] Salman A Baset and Henning Schulzrinne, “An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol”, Technical report Department of Computer Science Columbia University ,September 15, 2004 “http://arxiv.org/pdf/cs.NI/0412017 ,
5 Oct 2005

[35] Carlos Marco Arranz, “IP Telephony: Peer-to-peer versus SIP” Master Thesis, Department of Microelectronics and Information Technology Royal Institute of Technology (KTH) Sweden,
ftp://ftp.it.kth.se/Reports/DEGREE-PROJECT-REPORTS/ , 4 Sep 2005.

[36] David Partain, *“Skype –The operator as a bit-pipe*?” Ericsson Research, Linköping internal presentation, 21 Jan 2005

[37] Ericsson ABS internal documentation, CPEPS department, Ericsson, Linkoping ,
22 August 2005

Appendix A : Abbreviations and Acronyms

AS	Authentication Server
AAA	Accounting, Authorization, Authentication
AP	Access Point
BSSID	Basic Service Set Identity
DS	Distribution System
DSSS	Direct Sequence Spread Spectrum
ESSID	Extended Service Set Identity
IAPP	Inter Access Point Protocol
IEEE	Institute of Electrical and Electronic Engineers
IMS	IP multimedia Subsystem
IMT	IP Multimedia Subsystem Telephony
GTK	Group Transient Key
OFDM	Orthogonal Frequency Division Multiplexing
OTA	Over The Air
PMK	Pairwise Master Key
PTK	Pairwise Transient Key
PS	Power Save
MSN	Microsoft Network
QoS	Quality of Service
STA	WLAN Client Station

UDP	User Datagram Protocol
WiFi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPA IE	Wireless Protected Access Information Element
VoIP	Voice over Internet Protocol
VoWiFi	Voice over WiFi

COS/CCS 2006-2

www.kth.se