2010-04-01

MeshScan: a Fast and Efficient Handoff Scheme for IEEE 802.11 Wireless Mesh Networks

Yin Chen
Technological University Dublin, yin.chen@tudublin.ie

MeshScan - A Fast and Efficient Handoff Scheme for IEEE 802.11 Wireless Mesh Networks

By

Yin Chen

B.Eng.

A thesis submitted to the Dublin Institute of Technology

for the degree of

Master of Philosophy

Supervisor: Dr. Mark Davis

School of Electronic and Communications Engineering

April 2010

Abstract

As a next generation network solution, Wireless Mesh Networks (WMN) provides fast Internet access to a large area, which is from university campus to city scale. In order to provide an uninterrupted Internet experience to a mobile client, a process called handoff is required to maintain the network connection from one Mesh Node (MN) to another MN. Ideally, handoff should be completely transparent to mobile users. A critical application like VoIP will require a handoff capability that transfers a call from one mesh node (MN) to another in less than 50 msec. However the current IEEE 802.11 standards do not address the handoff well. Studies have revealed that standard handoff on IEEE 802.11 WLANs incurs a latency of the order of hundreds of milliseconds to several seconds. Moreover, the discovery step in the handoff process accounts for more than 99% of this latency.

The study addresses the latency in the discovery step by introducing an efficient and powerful client-side scan technique called MeshScan which replaces the discovery step with a unicast scan that transmits Authentication Request frames to potential MNs. A prototype of MeshScan has been developed based on the MadWifi WLAN driver on Linux operating systems. The feasibility of MeshScan to support fast handoff in WMNs has been demonstrated through extensive computer simulations and experiments under same given conditions. The results from the simulations and experiments show that the latency associated with handoff can be reduced from seconds to a few milliseconds by using the MeshScan technique. Furthermore, it is shown that MeshScan can continue to function effectively even under heavy traffic loads.

Declaration

I certify that this thesis which I now submit for examination for the award of _____, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

This thesis was prepared according to the regulation for postgraduate study by research of the Dublin Institute of Technology and has not been submitted in whole or in part for an award in any other Institute or University.

The work reported on this thesis conforms to the principles and requirements of the Institute's guidelines for ethics in research.

The Institute has permission to keep, to lend or to copy this thesis in whole or in part, on condition that any such use of the material of the thesis be duly acknowledged.

Signature_____Date_____

Acknowledgements

I would first like to thank my supervisor, Dr. Mark Davis for his invaluable advice and guidance during the course of my research. He has been extremely supportive and motivating throughout out the entire project particularly during difficult period.

I would like to thank all my colleagues at CNRI, past and present for their stimulating and often humorous conversations which provided me with a pleasant working atmosphere. In particular I owe a debt of gratitude to Dr. Karol Kowalik for giving up his valuable time and experience.

I would also like to express my thanks to Science Foundation Ireland for providing financial assistance.

Finally, I would like to thank my friends and family who have all contributed to this project in some form. This project would not have been possible without their support and understanding.

Table of Contents

List of Figures

List of Tables

Abbreviations and Acronyms

ACK	Acknowledgment
AID	Association ID
AP	Access Point
BPSK	Quadrature Phase Shift Keying
BSS	Basic Service Sets
BSSID	Basic Service Sets Identification
CoA	Care-of-Address
CRC	Cyclic Redundancy Check
CW	Contention Window
DA	Destination Address
DCF	Distributed Coordination Function
DIFS	DCF Interframe Space
DSSS	Direct Sequence Spread Spectrum
EMA	Exponential Moving Average
ESS	Extended Service Set
FA	Foreign Agent
FCS	Frame Check Sequence
FT	Fast BSS Transition mechanism

FHSS	Frequency Hopping Spread Spectrum
HA	Home Agent
HAL	Hardware Abstraction Layer
HR/DSSS	High Rate Direct Sequence Spread Spectrum
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and electronic Engineers
IR	Infrared
MAC	Medium Access Control
MCL	Mesh Connectivity Layer
MIMO	Multiple-Input and Multiple-Output
MN	Mesh Node
MSDU	MAC Service Data Unit
NIC	Network Interface Cards
NS2	Network Simulator Version 2
PLCP	Physical Layer Convergence Protocol
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time
SA	Source Address
STA	Station

TU	Time Unit
OFDM	Orthogonal Frequency Division Multiplexing
PCF	Point Coordination Function
PHY	Physical
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
VoIP	Voice over Internet Protocol
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMNs	Wireless Mesh Networks
Wi-Fi	Wireless Fidelity

1. Introduction

1.1 Challenge in Wireless Mesh Networks (WMNs)

Wireless Mesh Networks (WMNs) are a new architecture intended to provide a cost effective high-bandwidth network over a large coverage area. In recent years, WMNs have emerged as a promising solution to provide low cost access networks that extend Internet access and other networking services. A significant application for WMNs is VoIP. Wireless VoIP applications are beginning to emerge in the business market and IP Telephony revenues will more than double by 2013, compared to 2008, according to research from In-Stat [1]. Voice users are far more mobile than data users and will require a handoff capability that can transfer a call from one mesh node (MN) to another in less than 50 msec. [2-3]. Handoff introduces temporary variation in the delay – more appropriate to consider as jitter rather than delay. According to the IEEE 802.11 standard, the handoff process can take from a few hundred milliseconds to several seconds which is unacceptable [3]for VoIP users. Ideally, handoff should be completely transparent to mobile VoIP users, however the current IEEE 802.11 standard does not address this issue properly.

In July 2008, the IEEE published the final specification for the IEEE 802.11r-2008 standard [4], also known as Fast Basic Service Set Transition which is an amendment to the IEEE 802.11 standard that supports fast handoff between access points by introducing the Fast BSS Transition (FT) mechanism. The FT mechanism addresses two classes of network

infrastructures from a QoS perspective, but it still does not address the core questions of when and where a station (STA) will handoff to.

1.2 Overview of this study

Much of the work to date[5-7] in the area of handoff in IEEE 802.11 wireless networks has been concerned with essentially trying to duplicate the successful handoff mechanisms that already exist in cell phone networks when a mobile device roams between base stations. A cell phone handoff must be quick enough to support full-duplex voice communication without a perceivable gap in either voice stream.

Previous studies on seamless mobility in wireless mesh networks can be divided into two different categories: MAC layer handoff and network layer handoff. MAC layer handoff is often referred to as micro-mobility while network layer handoff is referred to as macro-mobility. Surveys addressing all of these areas are reviewed by Akyildiz et al. in [8] and [9]. Improvements have been shown in the previous studies [6-7, 10-14], but most of them are not specifically focused on WMNs or only address centrally controlled solutions which are expensive to deploy.

The objective of this thesis is to develop a MAC layer handoff scheme to provide fast handoff in WMNs. After a study and analysis of the IEEE 802.11 standard handoff procedure, the handoff process was divided into two phases: discovery phase (discovery latency) which is used to discover the available APs/MNs and the execution phase which includes two authentication and (re)association phases. A fast handoff management scheme

have been developed which called MeshScan [15] to provide a novel use of the channel scanning technique by employing open system authentication in both Passive Handoff and Active Handoff. MeshScan scheme comprises three steps: firstly a client device takes advantage of the WMNs architecture to maintain a list of active MNs (SmartList). Secondly MeshScan performs handoff when it receives a disassociation management frame from the serving MN or when the measured signal strength from the serving MN exceeds a given threshold. Thirdly when handoff is required, a client transmits Authentication Request frames to all MNs on the list instead of broadcasting Probe Request frames as is usually the case in an active scan in order to discover the available MNs. MeshScan handoff scheme may be implemented by upgrading the software on the client side, no hardware upgrade is required. NS2 simulations were used in order to verify the feasibility of the MeshScan scheme. A prototype of MeshScan was then implemented in the Linux-based Madwifi driver to demonstrate the effectiveness of our scheme through experiments [16]. In this project, it is assumed that the client has the SmartList preloaded in order to focus on the MeshScan MAC layer scan scheme. The SmartList is one of the key to the MeshScan, but again the project is aimed to verify and demonstrate MeshScan, as a new MAC layer scan technique.

The results presented in chapter 5 indicate that the latency associated with handoff can be reduced from seconds to a few milliseconds by using the MeshScan technique. The results from simulations and experiments show that 100% of handoff latencies were within 50 ms when there is no background load present. In the experiments, it was found that the average value of handoff latency is approximately 2.5 ms.

The performance of MeshScan was also analyzed under different network conditions where the number of MNs was increased from three to six nodes and under different background traffic loads of 10 Mbps, 15 Mbps, 20 Mbps, and 25 Mbps. The results show that MeshScan continued to successfully operate under different network conditions. For example, 75% of the handoff processes were completed within 50 ms under a 25 Mbps background load and where there were six MNs available to the STA. Also it was shown that the performance of MeshScan improves with the number of available MNs present.

1.3 Organization

This thesis is organized as follows.

Chapter 2 describes the reason why handoff is important for WMNs. It explains the concept of a WMN and the handoff procedure in the IEEE 802.11 standard with particular emphasis on the handoff related management frames at the MAC layer. This chapter also describes the various methods and protocols used for handoff.

Chapter 3 describes the proposed fast handoff scheme, MeshScan, in detail and describes the test-bed set up where the experiments were conducted. All the hardware used along with the protocols employed and programs developed to support the experiments undertaken are described in detail.

Chapter 4 describes the implementations of our proposed fast handoff scheme, MeshScan, in both NS2 and the Linux-based Madwifi driver.

Chapter 5 presents the results in 3 main sections. Each section is based on a different objective and follows a logical course of analysis. The first section is an analysis of the standard handoff process. The second is the simulation of our proposed MeshScan fast

handoff scheme in NS2. The third is the MeshScan prototype experiments carried out on the WLAN mesh test bed using the modified Madwifi driver.

Chapter 6 presents a summary of the main findings and conclusions arising from the experimental work carried out. It also suggests areas of further research.

1.4 Publication arising from this study

Two conference papers detailing the experimental findings of the thesis have been presented at two international conferences in 2009:

- MeshScan: Fast and Efficient Handoff in IEEE802.11 Mesh Networks

  The 7th ACM International Symposium on Mobility Management and Wireless Access (MobiWAC 2009), Tenerife, Canary Islands, Spain, October 2009.

- MeshScan: Performance of Passive Handoff and Active Handoff

  International Conference on Wireless Communication & Signal Processing (WCSP'09), Nanjing, China, November 2009.

2. Handoff Background

2.1 Fast Handoff

Handoff which is a significant challenge for wireless networks, especially for real-time applications, has not been well addressed in wireless network standards. Specifically, the handoff mechanism defined under the IEEE 802.11 standard adopts a hard handoff approach which requires that a station has to first break its connection with its old Access Point (AP) before connecting to a new AP. This can result in long handoff latencies. Researchers in this area have found that the handoff procedure in IEEE 802.11 WLANs typically takes hundreds of milliseconds and that almost 99% of the handoff delay arises from the process of searching for a new AP [17] to associate with.

With the growing demand for Wi-Fi devices running real time applications, e.g. Wi-Fi phones running VoIP applications such as Skype, the latency associated with handoff is becoming unacceptable. Therefore a new handoff scheme which provides for a fast handoff ability needs to be developed for the next generation of wireless networks which aims to complete the handoff process in less than 50 ms. Handoff introduces temporary variation in the delay that impacts on jitter. The target of 50 ms represents the recommended maximum jitter for acceptable VoIP quality [2-3], therefore the maximum handoff latency needs to be much less than 50 ms.

In this thesis, a practical fast handoff management scheme is presented called MeshScan for IEEE 802.11 WMNs which addresses the two key questions at the core of the handoff process: When should handoff be performed and which AP should the client associate with?

2.2 Wireless Mesh Network

The Wireless Mesh Network (WMN) [8] is a new architecture for wireless networking that incorporates existing and new radio technologies defining the overall structure, components and the inter-relationships between devices in the network. This means that mesh networking technology can be applied to practically any radio scheme, effectively allowing the best radio technology to fit the desired application: Personal area networks (i.e. Bluetooth, UWB), local area networks (i.e. Wi-Fi), and wide area networks (i.e. WiMax and cellular). WMN is set to become the predominant wireless network technology for next generation networks as it has many advantages compared to traditional wireless networks:

- Self Organising - each node determines the routing paths for itself, saving time and effort in administration.
- Wide Coverage - multi-hop networks extend the communications coverage around obstacles and over greater distances.
- Scalable – The size of the network may be increased by simply adding more nodes. The routing configuration is automatic, and there is no exponential rise in complexity as the network grows.
- Network Resilience - The self organising functions run continuously, so when changes occur in connections and reception (i.e. when the network topology changes) the mesh will automatically re-route around blockages in real time.
- Cost effective – Less cabling need compare to traditional wireless networks (ESS).

A WMN is a collection of wireless devices, typically operating as APs that utilize special routing algorithms to dynamically adapt to changes in the network. These changes may be triggered by factors such as environmental changes, movement of the Mesh Nodes

(MNs) or even failure of the MNs due to loss of electrical power. This project is particularly interested in WMNs based on the IEEE 802.11 standards where at least one MN has a wired Internet connection to act as a network gateway. The other MNs connect to the gateway MN in order to gain access to the backbone network. An example of an IEEE 802.11 mesh network is illustrated below in Figure.2.1 [18].

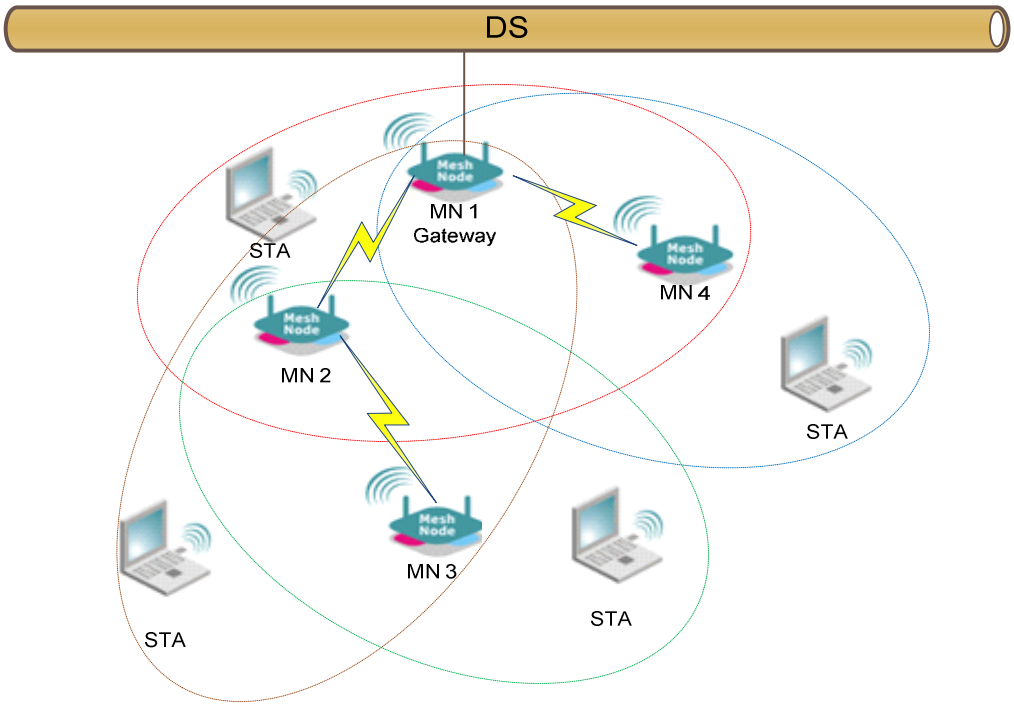


Figure 2.1: A Typical Wireless Mesh Network (WMN)

All of the MNs individually create Basic Service Sets (BSS) or hotspots and collectively create an Extended Service Set (ESS) or network coverage area. Usually, there will be a degree of overlap between the hotspots of different MNs. Mobile users roaming within the network coverage area will have to undergo handoff from one MN to another in order to preserve their network connectivity. The reason why user wants to use WMN

compare to telecom network (i.e. EDGE, 3G, upcoming 4G) for VoIP calls is because 802.11 based WMN is cheaper to use and in most of the case is free of charge.

2.3 802.11 Wireless Local Area Network Standards Overview

The IEEE Standard 802.11 was introduced in 1997 and has been regularly amended since then [19]. The IEEE 802.11 standard provides for a standards-based WLAN networking technology. The Wi-Fi Alliance [20] provided for inter-operability certification for WLAN equipment for different vendors based upon this standard. Three fundamental network building blocks are defined:

- Basic Service Set (BSS): Based upon the Infrastructure mode
- Independent Basic Service Set (IBSS): Based upon the Ad-Hoc mode
- Extended Service Set (ESS): Based upon the Infrastructure mode

The three different building blocks are shown in Figure 2.2 [19]

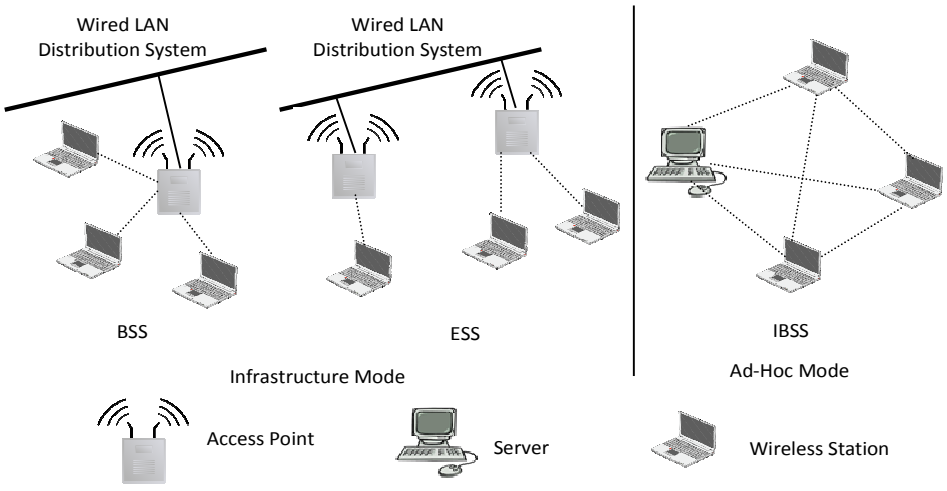


Figure 2.2: The BSS, ESS and IBSS Models Defined in the IEEE 802.11 WLAN Standard

There are two basic operation modes defined in the standard: Infrastructure mode and Ad Hoc mode.

- Infrastructure mode: In the infrastructure mode, the wireless network consists of at least one AP connected to the wired network infrastructure and a set of wireless end stations (STAs). All network communication takes place via the AP. The AP controls encryption on the network and may bridge or route the wireless traffic to a wired ethernet network (i.e. the Internet). APs that act as routers can also assign an IP address to a mobile device using DHCP services. APs can be compared to a base station used in cellular networks.

- Ad-Hoc Mode: The Ad-Hoc mode is a set of IEEE 802.11 STAs that communicate directly with each other without requiring the use of an AP. These networks are usually self-contained and do not have a connection to a wired network.

When using IEEE 802.11 radios to establish a WMN, generally two wireless interfaces (virtually or physically) are used in each MN. One interface works in ad-hoc mode to establish the WMN among all MNs and another interface works in infrastructure mode (i.e. it operates essentially as an AP) to provide a wireless connection to the end user.

List of IEEE 802.11 Standard are outland in Table 2.1

Table 2.1: IEEE 802.11 Standard List

Standard	Description
IEEE 802.11a	PHY Standard : 8 channels : 54 Mbps
IEEE 802.11b	PHY Standard : 3 channels : 11 Mbps
IEEE 802.11d	MAC Standard : operate in variable power levels
IEEE 802.11e	MAC Standard : QoS support
IEEE 802.11g	PHY Standard: 3 channels : OFDM and PBCC
IEEE 802.11h	Supplementary MAC Standard: TPC and DFS
IEEE 802.11i	Supplementary MAC Standard: Alternative WEP
IEEE 802.11r	Supplementary MAC Standard: Fast BSS transition
IEEE 802.11s	Supplementary MAC Standard: Mesh Networking

The IEEE 802.11 standard defines two network layers: Physical (PHY) and Medium Access Control (MAC) to provide for wireless connectivity for the STAs in a WLAN.

2.3.1 PHY Layer

The IEEE 802.11 WLAN system uses spread-spectrum wireless technology which is a wideband radio frequency technique. This technology is the foundation for wireless communications in the Industrial, Scientific & Medical (ISM) bands at 2.4 GHz [21-22] and 5 GHz [23]. Traditional radio communications focus on occupying as narrow a

frequency band as possible. Spread spectrum works by using mathematical functions to diffuse the signal power over a large range of frequencies. The receiver performs the inverse operation whereby the smeared out signal is reconstituted as a narrow band signal. This makes the data much less susceptible to electrical noise than conventional radio modulation techniques. Spread-spectrum is designed to trade off bandwidth efficiency for immunity to interference, integrity, and security. The original IEEE 802.11 standard defines three different types of PHYs, namely 2.4 GHz Frequency Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), and Infrared (IR). There are three more PHYs defined in subsequent amendments to the standard - High Rate Direct Sequence Spread Spectrum (HR/DSSS) [21-23], Orthogonal Frequency Division Multiplexing (OFDM) and Multiple-Input and Multiple-Output, (MIMO) [24].

This project considers WLANs based on OFDM technology as defined under the IEEE 802.11a standard which specifies a PHY for transmission at 5 GHz using OFDM modulation. In IEEE 802.11a, The OFDM system provides a WLAN with data payload communication capabilities of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s [23]. The support of transmitting and receiving at data rates of 6, 12, and 24 Mbit/s is mandatory. The OFDM system uses 52 subcarriers that are modulated using binary or quadrature phase shift keying (BPSK/QPSK), 16-quadrature amplitude modulation (QAM), or 64-QAM. Forward error correction coding (convolutional coding) is used with a coding rate of 1/2, 2/3, or 3/4 [23].

2.3.2 MAC Layer

The IEEE 802.11 standard specifies a common medium access control (MAC) Layer, which provides a variety of functions that support the operation of 802.11 based wireless LANs. The most important function of the MAC Layer is to manage and maintain

communications between IEEE 802.11 stations (Network Interface Cards (NIC) and Access Points (AP)) by coordinating access to a shared radio channel and utilizing protocols that enhance communications over a wireless medium. The IEEE 802.11 standard defines two forms of medium access, Distributed Coordination Function (DCF) and Point Coordination Function (PCF) [19].

The MAC address is a 48-bit unique identifier assigned to all NICs by the manufacturer for identification and is used in the media access control protocol sublayer. The MAC address is usually represented in hexadecimal format. For example, when all 48 bits are set to binary 1, it represents the broadcast MAC address which is FF:FF:FF:FF:FF:FF in hexadecimal format [19].

Some other functionalities (e.g. scanning, authentication, association) are also provided for in the MAC layer and these will be discussed further in the next section 2.3 which deals specifically with handoff in 802.11 WLAN.

2.3.3 MAC Management Frames in Handoff Process

Figure.2.3 [19] shows the generic 802.11 MAC management frame format and includes the size of each of the fields in octets or bytes.

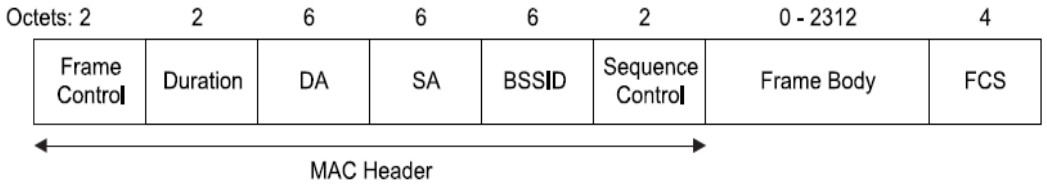


Figure 2.3: Generic 802.11 MAC Management Frame Format

Frame Control Field

Each MAC frame [19] begins with a 2-octets frame control field where defines the basic information of the frame. The format of frame control field is illustrated in Figure.2.4.[19]

The Type and Subtype fields indicate the type of frame (control, management or data) and WEP indicates frame body is encrypted according to the optional Wired Equivalent Privacy (WEP) algorithm.

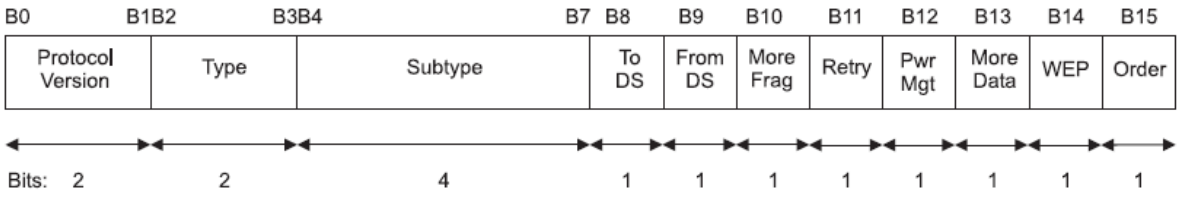


Figure 2.4: Frame Control Field

Duration/ID Field

The Duration field [19] follows the frame control field and its function depends on how bits 14 and 15 (WEP and Order shown in Figure 2.4) are set in power-save pool messages. This is the STA ID. In all other frames this is the duration value in microseconds used to set the NAV which is used by the MAC mechanism to control access to the medium.

DA (Destination Address)

The DA [19] is the destination MAC address of the management frame. If the frame is broadcasted, DA is set to the broadcast MAC address.

SA (Source Address)

The SA [19] is the MAC address of the station transmitting the MAC management frame.

BSSID

The BSSID field [19] is a 48-bit field of the same format as an IEEE 802 MAC address. This field uniquely identifies each BSS. The value of this field is the MAC address currently in use by the STA in the AP of the BSS in an infrastructure BSS.

Sequence Control Field

The sequence control field [19] is used to represent the order of different fragments belonging to the same frame and to recognise packet duplications. It consists of two sub-fields, fragment number and sequence number which define the frame and the number of the fragment in the frame.

Frame Body

The frame body [19] contains a MAC Service Data Unit (MSDU) or a fragment of an MSDU. This is also known as data field and its purpose is to move higher level payloads from STA to STA.

Frame Check Sequence (FCS)

The IEEE 802.11 frame ends with FCS field [19] which contains a 32-bit Cyclic Redundancy Check (CRC). The FCS allows a STA to check the integrity of received frames respectively.

2.3.3.1 Beacon Frame

The beacon frame [19] is one of the more important IEEE 802.11 WLAN management frames. Beacon frames are broadcasted periodically by the AP/MN in an infrastructure BSS to announce the presence of a WLAN. In IBSS networks, the transmission of beacon frames is distributed among the STAs.

The beacon interval indicates the time interval between beacon transmissions. The beacon interval is expressed in TU (Time Unit) [19] which is defined as a measurement of time equal to 1024μs in the IEEE 802.11 standard. It is a configurable parameter in the AP/MN and by default is configured as 100 TU (100 ms). Other information pertinent to the WLAN is transmitted in additional information fields and elements that are given in Table 2.2. [23]

Table 2.2: Beacon Frame Body

Order	Information	Notes
1	Timestamp	Timestamp for the current beacon frame transmitted
2	Beacon interval	AP/MN's configured Beacon interval parameter
3	Capability information	Currently capability information of AP/MN
4	SSID	The SSID element indicates the identity of an ESS or IBSS
5	Supported rates	AP/MN's NIC supported transmission rates
6	FH Parameter Set	The FH Parameter Set information element is present within Beacon frames generated by STAs using frequency-hopping PHYs
7	DS Parameter Set	The DS Parameter Set information element is present within Beacon frames generated by STAs using direct sequence PHYs.
8	CF Parameter Set	The CF Parameter Set information element is only present within Beacon frames generated by APs/MNs supporting a PCF.
9	IBSS Parameter Set	The IBSS Parameter Set information element is only present within Beacon frames generated by STAs in an IBSS.
10	TIM	The TIM information element is only present within Beacon frames generated by APs/MNs.

2.3.3.2 Probe Request Frame

A probe request frame [19]is sent from a STA when it requires information from another STA. The probe request frame body details are shown in Table 2.3 [23].

Table 2.3: Probe Request Frame Body

Order	Information
1	SSID
2	Supported Rates

2.3.3.3 Probe response Frame

A probe response frame [19] is sent by an AP after receiving a probe request frame and it contains capability information, supported data rates etc. The probe response frame contains the same information as the beacon frame, expect there is no TIM field in the probe response frame.

2.3.3.4 Authentication Frame

The authentication frame [19] is a management frame sent from a STA to the AP/MN that it wishes to authenticate with. The authentication process consists of the transmissions of two or four authentication frames which depends on the type of the authentication being implemented, i.e. open system or shared key respectively. The authentication frame body details are shown in Table 2.4.[23] Table 2.5 [23] shows the challenge test information in each status of authentication.

Table 2.4: Authentication Frame Body

Order	Information	Notes
1	Authentication algorithm number	Set to1 for open system and 2 for shared key
2	Authentication transaction sequence number	Two octets which indicates the current state of progress through a multistep transaction.
3	Status code	The status code information is reserved and set to 0 in certain Authentication frames as defined in Table 2.4.
4	Challenge text	The challenge text information is only present in certain Authentication frames as defined in Table 2.4.

Table 2.5: Presence of Challenge Text Information

Authentication algorithm	Authentication transaction sequence no.	Status code	Challenge text
Open System	1	Reserved	Not present
Open System	2	Status	Not present
Shared Key	1	Reserved	Not present
Shared Key	2	Status	Present
Shared Key	3	Reserved	Present
Shared Key	4	Status	Not present

2.3.3.5 Association Request Frame

The association request frame [19] is sent after a successful authentication from the STA to the AP/MN. The association request frame contains the information shown in Table 2.5. The listen interval field is used to indicate to the AP/MN how often a STA awakes to listen to beacon frames. The association request frame body detail are shown in Table 2.6 [23]

Table 2.6: Association Request Frame Body

Order	Information
1	Capability information
2	Listen interval
3	SSID
4	Supported rates

2.3.3.6 Reassociation Request Frame

The reassociation request frame [19] is similar to the association request frame, except that the reassociation request frame is trying to maintain an old connection or transfer the old connection with an old AP/MN to the new AP/MN. Therefore there is one more field in the reassociation request frame body than in the association request frame body. The reassociation request frame body details are shown in Table 2.7 [23]

Table 2.7: Reassociation Request Frame Body

Order	Information
1	Capability information
2	Listen interval
3	Current AP/MN's MAC Address
4	SSID
5	Supported rates

2.3.3.7 Association/Reassociation Response Frame

The Association/Reassociation response frame [19] is sent from the AP/MN to the STA after successfully receiving an association request frame. The Listen Interval field in the association request frame is used to indicate to the AP/MN how often an STA awakes to

listen to beacon frames. Association ID is assigned to STA by AP/MN after successful authentication. The association response frame body details are shown in Table 2.8 [23].

Table 2.8: Association/Reassociation Response Frame Body

Order	Information
1	Capability information
2	Listen interval
3	Association ID (AID)
4	Supported rates

2.3.3.8 Deauthentication Frame

The deauthentication frame [19] is sent to terminate a secure communication. Usually it is sent from an AP/MN to a STA after unsuccessful authentication between the AP/MN and STA. The deauthentication frame body contains just one field called the reason code which indicates the reason for the unsuccessful authentication. The deauthentication frame body shown in Table 2.9 [23] and the reason codes are defined in Table 2.10 [23].

Table 2.9: Deauthentication Frame Body

Order	**Information**
1	Reason code

Table 2.10: Reason Code

Reason Code	**Meaning**
0	Reserved
1	Unspecified reason
2	Previous authentication no longer valid
3	Deauthenticated because sending station is leaving (or has left) IBSS or ESS
4	Disassociated due to inactivity

5	Disassociated because AP is unable to handle all currently associated stations
6	Class 2 frame received from nonauthenticated station
7	Class 3 frame received from nonassociated station
8	Disassociated because sending station is leaving (or has left) BSS
9	Station requesting (re)association is not authenticated with responding station
10-65535	Reserved

2.3.3.9 Disassociation Frame

The disassociation frame [19] is sent from either a STA or AP/MN to terminate the current connection between the STA and AP/MN. An AP/MN sends a disassociation frame to a STA when it shuts down or reboots. A STA sends the disassociation frame to AP/MN before the STA is powered off. The AP/MN can then relinquish memory allocations and remove the STA from the association table. The disassociation frame body is same as the deauthentication frame body and contains a reason code field. The frame body details are shown in Table 2.11 [23]and the reason codes are defined in Table 2.10 [23].

Table 2.11: Disassociation Frame Body

Order	Information
1	Reason code

2.4 IEEE 802.11r Standard

In July 2008, the IEEE published the final specification for IEEE 802.11r-2008 [4], also known as Fast Basic Service Set Transition, which is an amendment to the 802.11 IEEE standard that supports fast handoff between APs. Specifically, it is intended to provide support for VoIP roaming on a Wi-Fi network with 802.1X authentication [25].

The new amendment to the IEEE 802.11 standard supports VoWiFi handoff between APs by introducing the Fast BSS Transition mechanism (FT). VoWiFi is a Wi-Fi based VoIP service which is designed to work on wireless devices such as a laptop or PDAs. The FT mechanism addresses two classes of network infrastructures from a QoS perspective: one where the transition-enabled AP is willing to provision QoS resources at reassociation time; and another where the AP needs to reserve the network infrastructure resources before transitioning. However, the FT mechanism does not address the question of when or to whom a STA will handoff to.

2.5 Handoff in 802.11 WMN

A handoff occurs in an IEEE 802.11WMN [26] when a mobile STA moves beyond the radio range of one MN and enters another coverage area at the MAC layer (e.g. where a STA moves from one BSS to another BSS or where both BSS are belonging to same ESS.) or when a mobile STA finds another AP/MN having a stronger beacon signal than the current one. During the handoff, management frames are exchanged between the STA and the MN. Consequently, there is a latency involved in the handoff process during which the STA is unable to send or receive traffic.

The original design of the IEEE 802.11 standards just considered the handoff signalling where the handoff procedure can be divided into three phases: discovery, authentication and association/reassociation [19].

2.5.1 Discovery

Discovery is the process used to allow the STA identify the available MNs within the RF coverage range. Two methods are defined in the IEEE 802.11 standard, namely passive scanning and active scanning.

In passive scanning, the STAs do not transmit any frames on the medium and instead wait and listen to each available channel for beacon frames which are broadcasted periodically by the MNs. Usually, the beacon frame transmission period is configured at 100 ms, which makes the timescale of MN discovery on the scale of a second since there are 11 available channels in United States and 13 available channels in Europe [27], and a STA must scan each channel in turn.

In active scanning, in order to determine whether a MN is operating on a particular channel, a STA periodically broadcasts probe request frames on a particular channel. When a MN receives a probe request, it replies with a probe response frame. As with passive scanning, the STA must scan all available channels in turn.

The time required (or latency) to scan one channel depends on two parameters: *MinChannelTime* and *MaxChannelTime*. Both of these are measured in steps of a TU which corresponds to an time interval of 1024 microseconds. They control the duration of scanning in each channel. *MinChannelTime* defines the minimum time required to scan one channel to guarantee the reception of a Probe Response frame. If a STA waits during *MinChannelTime* without receiving any Probe Response after broadcasting a Probe Request, it assumes that there is no AP available in this channel. On the other hand, *MaxChannelTime* represents the time required to guarantee the reception of the Probe Response frames from multiple APs available in the same channel. If a STA receives a Probe Response during *MinChannelTime* after broadcasting Probe Request, it must extend its waiting time to *MaxChannelTime* in case more Probe Responses might arrive in the same channel. The IEEE standard does not specify their values, however typical values are suggested from previous empirical studies [28-29] as shown below

- $MinchannelTime = DIFS + (aCW \min \times aSlotTime)$. where *aCWmin* is the maximum number of slots in minimum contention window, and *aSlotTime* is the length of a slot. *DIFS* = 50μsec, *aCWmin* = 31μs and *aSlotTime* which is defined in the standard to be 20μsec in 802.11 b/g and 9μs in 802.11a. According to the analysis carried out in [28], it suggests an ideal value of this parameter lies between 1 ms [28] and 7 ms [29].
- *MaxChannelTime* is suggested to be set to approximately 11 ms [28-29]

Another issue in the discovery phase is the channel switching delay. This overhead is a characteristic of the network interface design and reflects the time required to switch to a new frequency, resynchronize and start demodulating packets. Channel switching delay varies considerably across implementations from a maximum of 19 ms (12 ms to switch and 7 ms to resynchronize) for Intersil Prism2-based NICs to just over 5 ms for Atheros 5212-based NICs according to previous study [7]. Since this cost is per channel it adds considerable delay to the overall scanning process.

2.5.2 Authentication

Authentication is the phase used to verify the identities involved between a MN and a STA and to bring the wireless link up to the assumed physical standards of a wired link. The IEEE 802.11 standard defines two authentication algorithms: open system and shared key authentication.

Open system authentication is the default authentication algorithm and any STA that requests authentication with this algorithm may become authenticated if the MN uses open system authentication. Open system authentication involves a two step authentication

transaction. The first step is the identity assertion and request for authentication and the second step is the authentication result. If the result is successful, the STA is mutually authenticated with MN. The minimum time required for authentication is two RTTs (Round Trip Times) for open system authentication. RTT is the time corresponding to the transmission time of a probe request frame and an ACK response frame between two nodes [30]. Four timestamps are required to calculate the RTT using Equation (2.1), due to the packet process delay.s

$$RTT = (T_{21} - T_{11}) + (T_{12} - T_{22}) \quad (2.1)$$

This study assumes that T_{11} is the timestamp of the probe request frame that is transmitted from Node A, T_{21} is the time that the request frame from Node A is received by Node B, T_{22} and T_{21} are similar to T_{11} and T_{21}, as shown in Figure 2.5. RTT depends on a number of factors that includes the network load, interference and contention.

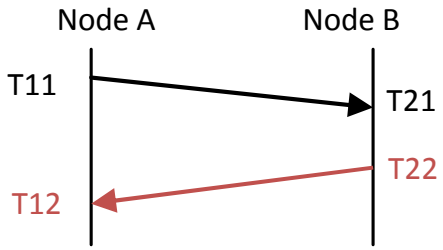


Figure 2.5: Round Trip Time

Shared key authentication is the same as open system authentication which allows any STAs to establish a link connection, but only a STA who knows the shared secret key can receive encrypted data. Shared key authentication involves a four step authentication

transaction. The first step is identity assertion and request for authentication and the second step is a challenge text sent back to the STA, the third step requires the STA to send encrypted challenge text back to the MN, and the final step is the authentication result. If the encrypted challenge text is correct, the STA is mutually authenticated with MN. The minimum time required for shared key authentication is four RTTs.

2.5.3 Association/Reassociation

Association is the process that follows after a successful authentication where the STA is assigned a proper association identity and the required resources by the new MN. Reassociation is a service that is invoked to move a current association from one MN to another. This keeps the DS informed of the current mapping between MN and STA as the station moves from BSS to BSS within an ESS. The minimum time required for both association and reassociation is four RTTs. Association/Reassociation represents the end of the handoff process in MAC layer.

2.5.4 Handoff Procedure and Delay

Figure 2.6 and Figure 2.7 [31-32] illustrate the basic handoff procedures for both passive scanning and active scanning respectively. The two procedures show the relevant delay associated with each step in the handoff procedure. The overall delay is the summation of scanning delay, authentication delay and association/reassociation delay.

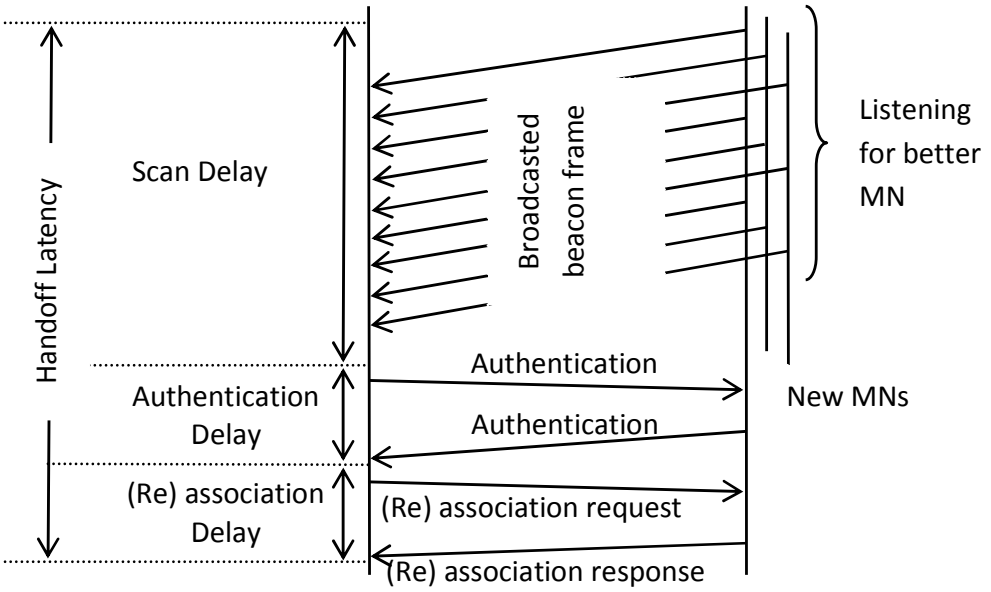


Figure 2.6: The Handoff Procedure in Passive Scanning

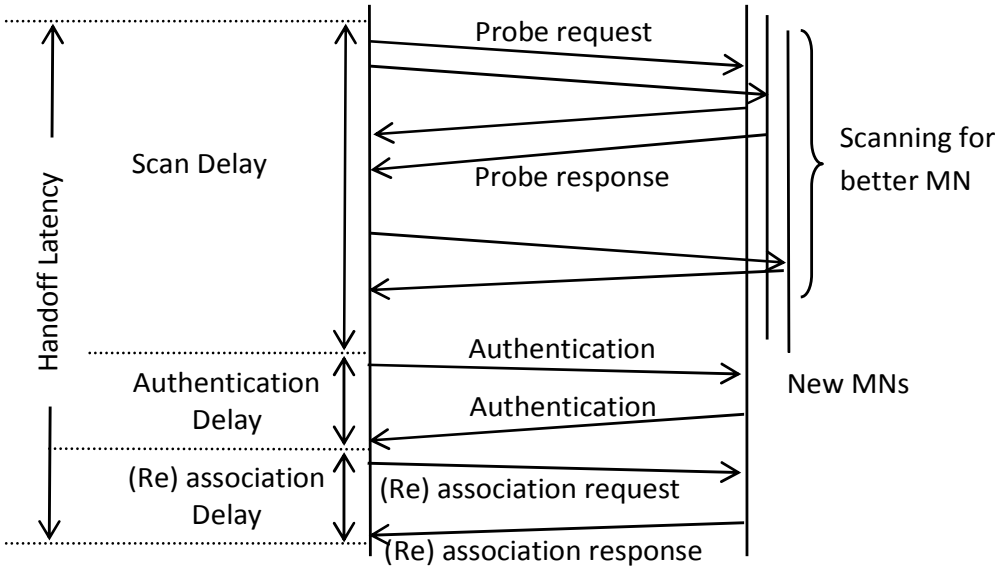


Figure 2.7: The Handoff Procedure in Active Scanning

2.6 Related Work

2.6.1 Wireless Mesh Networks

There has been a considerable amount of work carried out on wireless peer based networking. One of the first commercial mesh networks was Metricom's Ricochet network [33] in the mid-90s. Ricochet nodes automatically route client traffic through half-duplex wireless hops until reaching a cable connection.

When the IEEE 802.11 standard was ratified in the late-90s, other mesh networks started to emerged. One of these is the MIT Roofnet [34-35] project where tens of MNs with roof mounted antennas formed a mesh around campus. Roofnet's emphasis is more on route maintainability and optimization than on handing off a client's connection. Many other community and commercial mesh network implementations also exist, such as Rice University TAPS in Houston [36] and Urbana-Champaign Community Wireless Project [37].

Microsoft Research has also done notable work in the area of mesh networks. Their Mesh Connectivity Layer (MCL) [38] creates a wireless mesh network between Windows clients. Their approach focuses on efficient routing protocols along with the unique supported for multiple radios on each node. Adya, Bahl, Wolman, and Zhou have shown [39] that using multiple radios on a mesh node combined with smart routing algorithms [40] will dramatically improve the throughput of a wireless mesh network. Their work necessitates a specific network driver on all mesh network participants, including the clients.

Existing experimental wireless mesh testbeds that support client mobility include MeshCluster [41] and iMesh [42], both of which work with mobile clients in the infrastructure mode. MeshCluster, which uses MIP for MAC layer handoff, shows a latency of about 700 ms due to the delay incurred during access point re-association and MIP registration. iMesh also offers MAC layer handoff using regular route updates or Mobile IP. Using layer-2 handoff triggers (no moving client), handoff latency in iMesh takes 50-100 ms. The approach was later used in a more realistic environment for improving VoIP performance in mesh networks, with similar results [43]. SMesh [44-45] provides IEEE 802.11 link-layer and network-layer fast handoff by working in ad-hoc (IBSS) mode, controlling handoff from the mesh infrastructure, and using multicast to send data through multiple paths to the mobile client to deal with incomplete knowledge and unpredictable moving patterns.

2.6.2 Network Layer Handoff

Two general approaches for supporting network layer handoff are Mobile IP (MIP) [46]and Mobile NAT [47]. In MIP, a client binds to an IP address at the Home Agent (HA). As the mobile client moves to a different access point or domain, it receives a Care-of-Address (CoA) from a Foreign Agent (FA). The mobile client then registers its new CoA with its HA and data is then tunnelled through the HA.

In Mobile NAT, a client receives two IP addresses through DHCP: a binding address for the network stack, and a routing address that will be visible in the network. As the mobile client moves to a different domain, the client may receive a new routing address. However, as end-to-end connections were initiated from the IP address of the network stack (which remains the same), existing connections will be maintained. This approach requires

modifying the mobile client network stack to be aware of the protocol and also requires changes to the standard DHCP protocol.

Many reactive approaches have been proposed to address Internet connectivity in wireless ad-hoc networks [48-52]. Some of them provide good connectivity while paying the cost of a fairly high overhead due to periodically advertisements from Foreign Agents. Others use a reactive approach and broadcast advertisements to find Foreign Agents on demand which adjusts slowly. A hybrid approach that achieves the same connectivity as in pro-active protocols but with less overhead was proposed in [53]. These schemes usually share similarities with Mobile-IP and although they are suitable for ad-hoc networks, they do not perform well in wireless mesh networks. Backbone nodes in a mesh network are stationary, as opposed to the nodes in ad-hoc networks, leaving more opportunity for more efficient protocols that exploit the relative stability of the mesh nodes.

Two well known general approaches to network layer handoff are Cellular IP [54] and Hawaii [55]. A comparison is presented in [56]. In Hawaii, or Handoff-Aware Wireless Access Internet Infrastructure, messages are exchanged between the old gateway and the new gateway for forwarding packets. Cellular IP establishes routes based on traffic from the client and handoff takes place when a cross-over router is reached. However, applications like Push-to-Talk [57] may require packets to be sent to mobile clients that are only receiving traffic. In addition, these approaches rely on clients initiating the handoff process and do not address the link level handoff delay present in IEEE 802.11 networks when clients reassociate with another AP. Other approaches to network layer handoff, such as TMIP [58] and [59], improve handoff latency in IEEE 802.11 networks but do not

overcome these limitations. Other general approaches such as IDMP [60], SMIP [61], and HMIP [62] focus on hierarchy to reduce the global signalling load to improve scalability.

In [63], Caceres and Padmanabhan propose the use of gratuitous ARP messages to achieve transparency in the wired infrastructure during handoffs. In their approach, mobile clients initiate the handoff themselves and the APs send gratuitous ARPs to their upstream routers to create the illusion that mobile clients are always connected to the wired network. The approach requires all APs to be directly connected to the same wired ethernet network.

Helmy, Jaseemuddin, and Bhaskara show in [64] how fast handoff can be achieved in wireless networks by requiring mobile clients to explicitly join a multicast group to which packets are multicast-tunnelled through the infrastructure. Multicast during handoff, referred to as simulcast, is also used during handoff in SMIP [61]. In a different approach, Forte and Schulzrinne [65] propose a scheme where clients collaborate in multicast groups with each other clients in their vicinity to share useful information about the network and improve handoff performance.

2.6.3 MAC Layer Handoff

Cell networks achieve seamless handoff by sharing information between base stations about a given mobile device. This session data is used for routing and is updated whenever a phone switches cells [66-67]. The IEEE 802.11 standard lacks the handoff mechanisms available in today's cell network protocols.

Mishra, Shin, and Arbaugh [29] analyzed the link-level handoff performance in current IEEE 802.11 hardware. Approximately 90%of a handoff delay is attributable to the client

adapter scanning for its next AP. Their experiments also illustrate that the practical handoff delay can vary widely depending on the vendors used for the client network card and the AP. Vatn [68] investigated the latency effects of a wireless handoff on voice traffic. His conclusions echo those of Shin and Arbaugh in that the handoff latency can vary widely depending on the hardware vendor used.

Ramani and Savage [7] has demonstrated that a quick link-level handoff is possible on IEEE 802.11 networks when the client monitors the signal quality of APs and uses a fast scanning mechanism to listen to all APs in range to choose the best one. Their SyncScan system has achieved an impressive handoff as low as 5 ms. Other similar approaches such as shared beacon channel [69], dual re-authentication scheme[14], and multiple Wireless Network Interface Cards (Multi-WNICs) AP [70]. These hardware augmentation approaches have a deployment difficulty due to its high overhead and power consumption concerns.

The IEEE has also been working on IEEE 802.11 standard for handoff. Intel has carried out a performance study on the IEEE 802.11r fast BSS transition which reduces handoff delays associated with 802.1X authentication by shortening the time it takes to re-establish connectivity after a client transitions from one IEEE 802.11 AP to another while roaming [4, 25]. They have conducted experimental work on a test bed which included two APs and a STA. The STA was moving between the two APs with traffic comprising a two-way voice over IP over WLAN RTP traffic using a 20 ms codec between the STA and associated AP. The results have shown that the IEEE 802.11r standard results in an improved handoff latency of approximately 40 ms [71].

There are few wireless conditions that will impact on handoff delay from milliseconds to seconds. Point 1, 2 and 3 can be controlled by testbed setup to allow repeatable simulations and experiments. Point 4 is environment depend which varies over time (hard to control).

1. Operation mode (i.e. a/b/g): different number channel that need to be scanned. There are maximum 52 channels available in mode a and 13 channels in mode b/g.
2. The IEEE 802.11 interface parameters which controls the time requires to access the medium, such as *SlotTime, SIFS, CWmin,* and *CWmax* etc.
3. Heavy network traffic: it introduces high level of contention which increases the time require to access the channel- delay in management frame transmission.
4. Interference: it results to retransmit the management frame i.e. (re)authentication frame, (re)association frame, disassociation frame.

In this project, major network parameters were set as following in Table 2.12 for both simulations and experiments. Minor network parameters were set according to the particular simulations and experiments and will list in following chapters.

Table 2.12 Major Network Parameters

Operation Mode	802.11 a
Operation Channel	Channel 60 (5.32 GHz)
SIFS	10μs
SlotTime	20μs
CWmin	31
CWmax	1023
Data Rate	11Mbps

2.6.4 MeshScan

These fast handoff solutions discussed above are either centrally controlled handoff solutions for WMNs or not specifically focussed on handoff in WMNs. A new fast handoff management concept has been developed by the author for WMNs to address the latency in the discovery step by introducing an efficient and powerful client-side technique brand name called MeshScan [15-16] to solve two core problems in handoff, namely when the handoff should be performed and which MN that the client should associate with.

The basic idea of MeshScan is to reduce the discovery latency in order to allow the handoff process to take place in much less than 50ms. MeshScan scheme comprises three steps: firstly a client device takes advantage of the WMNs architecture to maintain a list of active MNs (SmartList which is considered as a given in this work and preloaded on client side in the experiments). Secondly MeshScan performs handoff when it receives a disassociation management frame from the serving MN or when the measured signal strength from the serving MN exceeds a given threshold. Thirdly when handoff is required, a client transmits Authentication Request frames to all MNs on the list instead of broadcasting Probe Request frames as is usually the case in an active scan in order to discover the available MNs. MeshScan handoff scheme may be implemented by upgrading the software on the client side, no hardware upgrade is required. Because MeshScan addresses fast handoff in the discovery phase and leaves the execution phase to operate as defined in the IEEE 802.11 standards, MeshScan is compatible with the recent IEEE 802.11r standard.

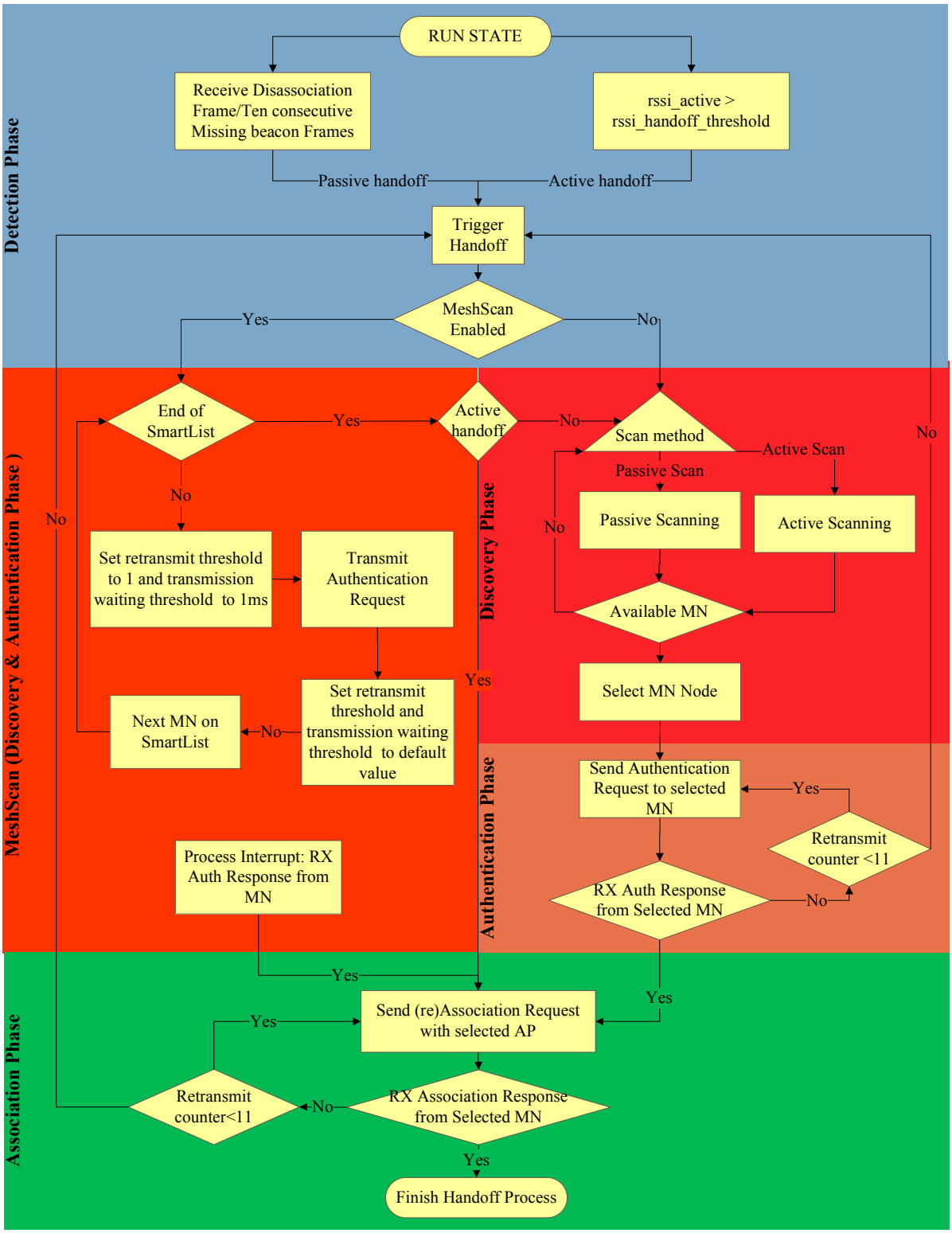


Figure 2.8: Handoff Procedure in modified Madwifi Driver with MeshScan scheme

A prototype of MeshScan has been developed to provide a novel use of the channel scanning technique by employing open system authentication in both Passive Handoff and Active Handoff, based on the Madwifi WLAN driver on Linux operating systems as shown above in Figure 2.8. The feasibility of MeshScan to significantly support fast handoff in WMNs has been demonstrated through extensive computer simulations and experiments in same network configuration (802.11 a mode, Channel 60, Open Authentication Key, 11Mbps data rate). From these simulations and experiments results, the handoff latency was up to seconds when standard handoff procedure was called, but when MeshScan technique was used the latency associated with handoff can be reduced to a few milliseconds in same network conditions.

The MeshScan scheme is a WMN centric solution, but can also be applied to more general IEEE 802.11 networks. The limitation of the MeshScan is that MeshScan is a new scanning technique relying on a list of available MNs. MeshScan does not generate this list, in this work the list of available is preloaded to the client side in order to perform MeshScan and this work is focused on to study the performance of new MeshScan scanning technique.

2.7 Chapter Summary

This chapter has discussed what handoff is and why fast handoff is important to WMNs and also outlines some fundamental aspects of the operation of IEEE 802.11 WLANs and WMNs. A detailed description of the handoff procedure in IEEE 802.11 standards was given that included the handoff related management frames and scan techniques. The chapter ended with discussion of related works and how they compare with our proposed

MeshScan fast handoff solution. The following chapters will further outline the technical details of the MeshScan scheme and its implementation, as well as an analysis of its performance.

3. MeshScan Validation Details

In this section, a detailed description of the MeshScan simulation and experimental analysis procedures will be presented in three sections. This chapter will also describe the experimental test bed and all relevant software tools used.

As discussed in the introduction chapter, this project focuses on reducing the MAC layer handoff latency to less than 50 ms to allow time critical applications to continue to operate during the handoff process. This study comprises three major sections: The first section investigates the 802.11 standard handoff process and analyzes the latencies related to the three phases associated with the handoff process. This section also describes the development of the MeshScan fast handoff scheme presented in this work. The second section is concerned with a computer simulation of the proposed MeshScan fast handoff scheme in order to compare its performance with that of the IEEE 802.11 standard handoff scheme. Finally, the MeshScan scheme was implemented in Linux to facilitate an experimental comparison between the handoff performance under MeshScan and the IEEE 802.11 standard. Different approaches were used in each section and they are described as follows.

3.1 Handoff Analysis

3.1.1 Objective

The objective in this section is to analyze the three phases that comprise the handoff process under the IEEE 802.11 standard. The latency associated with each of the three phases is analyzed and the MeshScan fast handoff scheme is introduced to address the needs of time critical applications.

3.1.2 Approaches of Study

As discussed in chapter 2, the IEEE 802.11 standard handoff process can be divided into three phases: discovery, authentication, and (re)association. Different approaches are used in order to carry out fast and reliable study on handoff which includes mathematical modelling, computer simulation, and network experiments.

3.1.2.1 Discovery Phase

Two scan methods can be used in the discovery phase: passive scanning and active scanning. Three approaches are used here to study the latency of the discovery phase: mathematical modelling, computer simulation, and network experiments.

3.1.2.1.1 Passive Scanning

As described earlier in the chapter 2, passive scanning latency depends on the beacon interval which is set at AP/MN side and can be calculated theoretically according to equation 3.1 where *latency_passive_scanning* is the total latency required for the passive scan method, *available_channel* is the number of channels required to be scanned during passive scanning, and *beacon_interval* is the time interval between successive beacon frames [72].

$$latency\_passive\_scanning = available\_channel \times beacon\_interval \qquad (3.1)$$

Equation 3.1 shows that the beacon interval is a key factor that determines passive scan latency. The passive scanning latency can be reduced by changing the beacon interval, but the bandwidth used for the transmission of beacon frames will increase correspondingly. In order to study the bandwidth usage for different beacon intervals a mathematical model was

developed where the following equations and variables are described in Table 3.1. Using equations 3.2 to 3.5, the bandwidth consumed by the beacon frames can be calculated as the *beacon_interval* varies. A threshold for *beacon_interval* can be chosen to avoid significantly increasing the bandwidth consumed by the beacon frames.

Table 3.1: Components List for Equation 3.2 to 3.5

$Beacon_{MAC}$	MAC frame size for beacon
Beacon_Frame_Size	The size of beacon frame body in bytes
FCS	Size of FCS field in bytes
MAC_Header	Size of MAC header field in bytes
PLCP	Transmission time for PLCP Preamble field
PLCP_Header	Transmission time for PLCP header field
Aver_backoff	The average backoff time
DIFS	The time the medium has to be idle before activating the backoff counter
Beacon_interval	Time interval between beacon transmissions
T_B	Beacon transmission time
R_B	Number of beacon per second
C_B	Capacity used by the beacon frames
TX _ Rate	Transmission Rate
TU	Time Unit

$$Beacon_{MAC} = (Beacon\_Frame\_Size + FCS + MAC\_Header) \times 8 \quad (3.2)$$

$$T_B = PLCP + PLCP\_Header + \frac{Beacon_{MAC}}{TX\_Rate} + Aver\_Backoff + DIFS \quad (3.3)$$

$$R_B = \frac{1}{TU \times Beacon\_interval} \quad (3.4)$$

$$C_B = T_B \times R_B \quad (3.5)$$

3.1.2.1.2 Active Scanning

As discussed earlier in chapter 2, active scanning is based on two parameters: *MinChannelTime* and *MaxChannelTime*. The IEEE 802.11 standard does not mandate a value for them, so there is a large variation in active scanning latency among different manufacturers [29]. Therefore a number of experimental tests were carried out to analyze the active scanning latency using the CNRI Mesh testbed [73] which will be discussed in detail later in this chapter.

The experimental testbed includes between three and six MNs, one STA and one WMN Controller which remotely controls the wireless interfaces on three MNs. All MNs operate using 802.11a on channel 60 (i.e. in the 5 GHz ISM band) and use open system key authentication. The STA was running Fedora 10 with a 2.6.27.24-170.2.68 kernel on a Pentium(R) PC platform (Dual-Core CPU E5200 2.5GHz, 1GB RAM) with an Atheros AR5212-based wireless interface. By default, the Madwifi driver uses active scanning during handoff. The Madwifi driver required some modifications which involved the addition of an event manager for logging the timestamped management frames (derived from the Linux kernel layer) which allows for determining the latency in each handoff

phase. An automated script runs on the WMN controller to force STA handoff among the three MNs by turning the MN radio interfaces on and off. The experimental testbed is shown in Figure 3.1 below. The STA and three MNs have fixed positions to permit repeated experimental work. The location of the STA was covered by all three MNs to allow the STA to handoff between each of the MNs. More than 1000 handoff process were conducted to provide for a large sample population. A kernel system log was generated for each experiment by the madwifi driver. The system log file contains a timestamp for every handoff related management frame and a perl script was used to analyze the output file and to calculate the handoff latency based on disassociation and association frames.

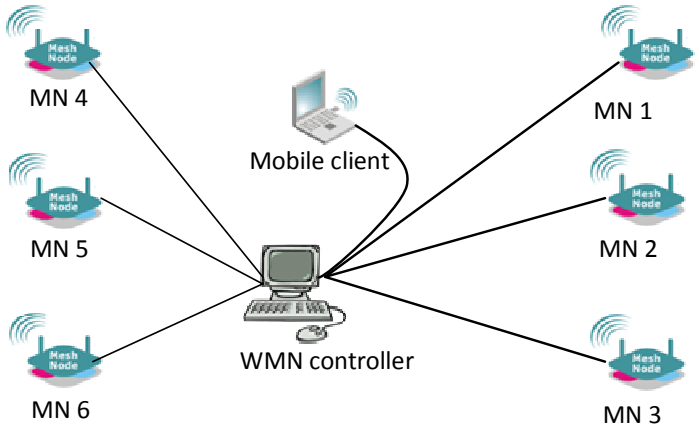


Figure 3.1: Active Scanning Experimental Setup

3.1.2.1.3 Authentication Phase and (Re)association Phase

As discussed earlier in chapter 2, the authentication phase and (re)association phase are the two phases whereby the STA exchanges frames with the AP/MN to establish the connection. This project assumes that open system authentication only is used for all MNs in both simulation and experimental works. Both simulation and experimental tests were carried out to determine the latency for both the authentication phase and (re)association

phase. The same experimental testbed setup was used as for the active scanning (Figure 3.1). The open source Linux based network simulator (NS2) was used as the simulation tool. The simulation scenario included three MNs and a STA as shown in Figure 3.2.

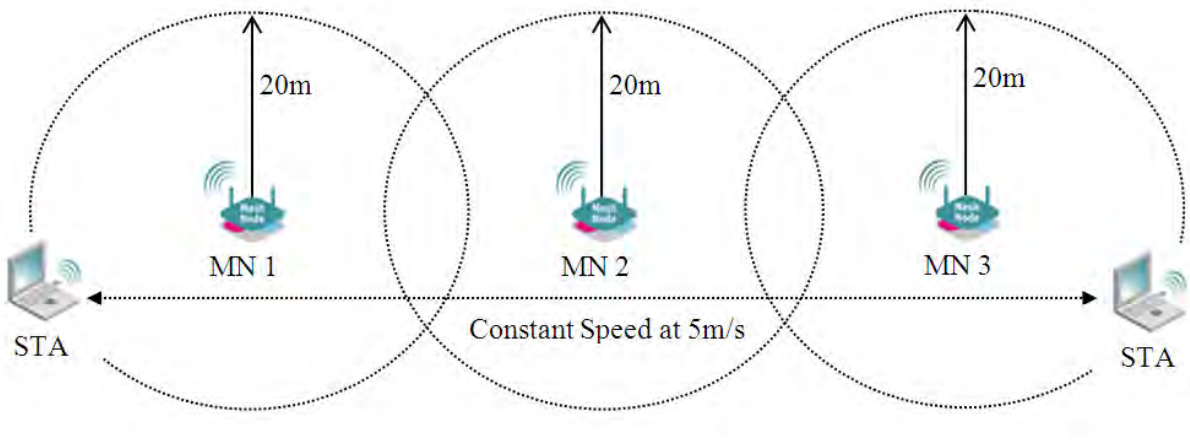


Figure 3.2: Basic Handoff Simulation Scenario

All MNs and the STA operate using the 802.11a mode on channel 60 and use open system key authentication. All MNs have a 20 meter coverage radius and the coverage areas are overlapped as shown in Figure 3.2. The STA was moving back and forth among the three MNs at a constant speed at 5 m/s for two thousand times to allow for a reliable statistical analysis to be performed. The same simulation was repeated by using different beacon frame intervals at 100 ms, 90 ms, 80 ms, 70 ms and 60 ms. An output file was generated by NS2 after each simulation and stored on local server. The output file contains STA movement (speed and direction), data packets flow information (source, destination, transmitting time and receiving time etc.) and IEEE 802.11 management frame information (source, destination, transmitting time and receiving time etc.).

3.2 MeshScan handoff Schemes

Having analyzed the latency results from the previous section, it was found that the discovery phase was responsible for over 99% [17] of the total latency of the overall

handoff process. The authentication phase latency and (re)association phase latency could not be improved unless the original IEEE 802.11 standard were to be changed. This is the subject of the recent IEEE 802.11r amendment to the standard. As described in chapter 2, the new IEEE 802.11r standard specifies a solution to two classes of network infrastructures from a QoS perspective. However, it does not address the issue of when or where a STA will handoff. In other words, the IEEE 802.11r standard only seeks to reduce the latency which is introduced in both the authentication and (re)association phases. It does not seek to improve the latency case associated with the discovery phase.

Therefore the handoff process was divided into two phases: discovery phase (discovery latency) which is used to discover the available APs/MNs and the execution phase which includes two authentication and (re)association phases. A fast handoff scheme have been developed called MeshScan which is specifically focused on WMN applications. The basic idea of MeshScan is to reduce the discovery latency in order to allow the handoff process take place in less than 50 ms. Because MeshScan addresses fast handoff in the discovery phase and leaves the execution phase to operate as defined in the IEEE 802.11 standards, MeshScan is compatible with the recent IEEE 802.11r standard.

MeshScan is based on preloaded a list of all available MNs called a MN list. The MN list can be learnt or cached on the STA. When handoff is required, the STA performs a unicast scan by transmitting Authentication Request frames to each of the MNs on the MN list to discover the next MN for handoff. This avoids having to use either passive or active scanning to discover the available MNs to associate with. Both computer simulations and experiments were used to validate the operation of MeshScan and to compare MeshScan with standard scan techniques.[16]

Table 3.2: MeshScan Latency

	Passive Handoff	Active Handoff
Handoff Triggering Latency	0	$\frac{1}{2}RTT$
Authentication Latency	$m\times(\frac{1}{2}RTT)+RTT$	$m\times(\frac{1}{2}RTT)+RTT$
Association Latency	$2RTT$	$2RTT$

In terms of the algorithmic delay associated with MeshScan in the Madwifi driver and assuming at least one MN is available. Table 3.2 shows how the handoff latency was measured and the latency of each step in both passive and active handoff. Passive handoff triggers when a disassociation frame is received from current associated MN. Active handoff triggers when a disassociation frame is sent to current associated MN, no ACK is expected. Equation (3.6) [16] applies to passive handoff and equation (3.7) [16] applies to active handoff where M is the number of Authentication Request frames transmitted and *ChannelSwitchTime* is the time required by the NIC to switch from one channel to another. In the best case scenario the first MN from the SmartList is the next MN to re-associate with, so the delay is $3\frac{1}{2}RTT$ in Passive Handoff and $4RTT$ in Active Handoff. The worst case will be where there is no available MN and the mobile client must carry out active scanning.

$$M\times(\frac{1}{2}RTT)+3RTT+(M-1)\times ChannelSwitchTime \tag{3.6}$$

$$M\times(\frac{1}{2}RTT)+3\frac{1}{2}RTT+(M-1)\times ChannelSwitchTime \tag{3.7}$$

3.2.1 MeshScan Simulations

Different simulation scenarios were used to verify the feasibility of MeshScan and to compare its latency with that of the standard IEEE 802.11 handoff latency. Each scenario involved only one mobile STA, but different numbers of MNs and different topologies were used depending on the number of MNs used in simulation. NS2 was used as the simulation tool and a modification was required to simulate MeshScan. This modification of NS2 will be discussed in the next chapter.

In all simulation scenarios, all the MNs and the STA operate using the 802.11a mode on channel 60 and open system key authentication was used. All MNs have a 20 meter coverage radius and the coverage areas overlap. STA was moving back and forth among the MNs at a constant speed at 5m/s for two thousand times for each scenario in order to obtain reliable simulation results. This particular movement pattern was chosen for a number of reasons. Firstly, it allowed NS2 to run the simulations without crashing. Secondly different contention areas (low, medium and high) were introduced into the simulation scenario. Different MN topologies were used in each scenario to ensure that the coverage areas of all the MNs overlapped with each other, so as to ensure that the mobile STA would be able to associate with anyone of them. The same simulation was repeated using different beacon frame interval at 100 ms, 90 ms, 80 ms, 70 ms, and 60 ms. The MN topology and STA path used in each scenario are shown in the following figures from Figure 3.3 to Figure 3.6.

Figure 3.3 shows the MN topology and the movement of the mobile STA when the simulation scenario involved three MNs.

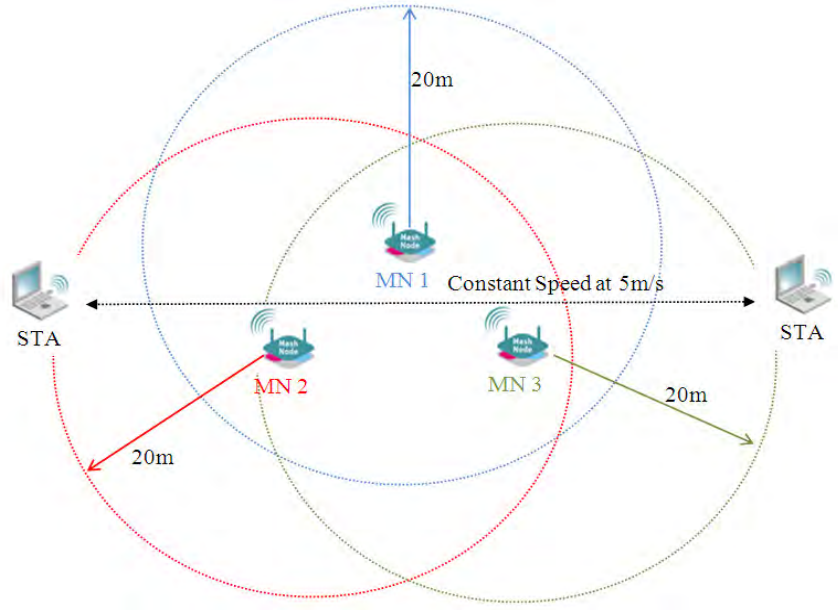


Figure 3.3: Three MNs Handoff Simulation Scenario

Figure 3.4 shows the MN topology and the movement of the mobile STA when the simulation scenario involved four MNs.

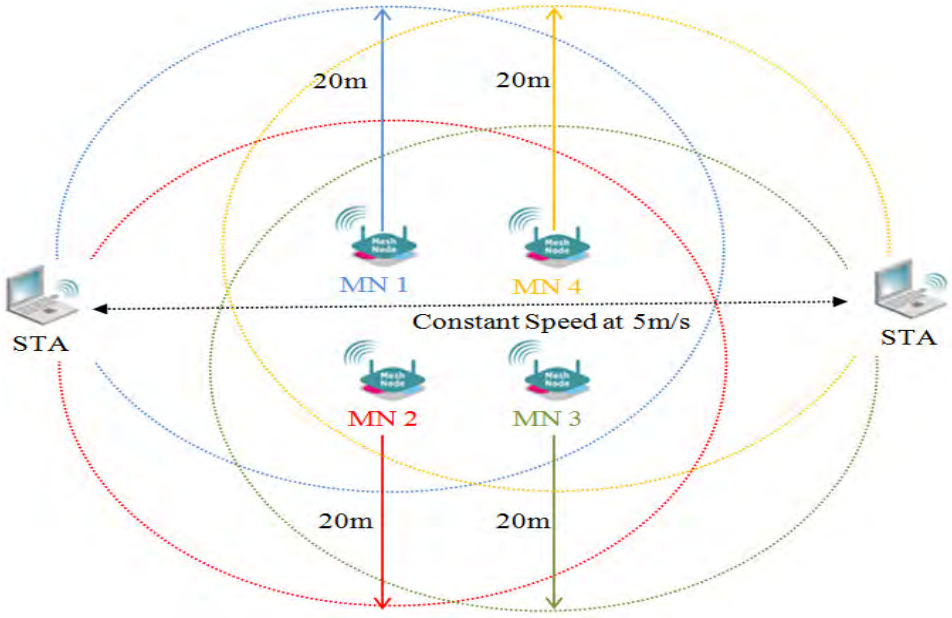


Figure 3.4: Four MNs Handoff Simulation Scenario

Figure 3.5 shows the MN topology and the movement of the mobile STA when the simulation scenario involved five MNs.

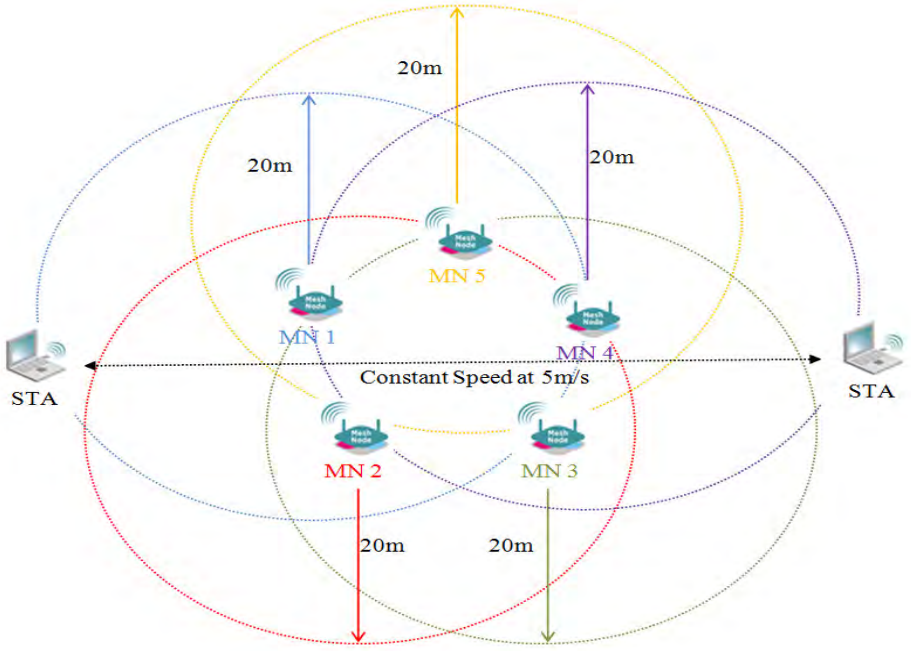


Figure 3.5: Five MNs Handoff Simulation Scenario

Figure 3.6 shows the MN topology and the movement of the mobile STA when the simulation scenario involved six MNs.

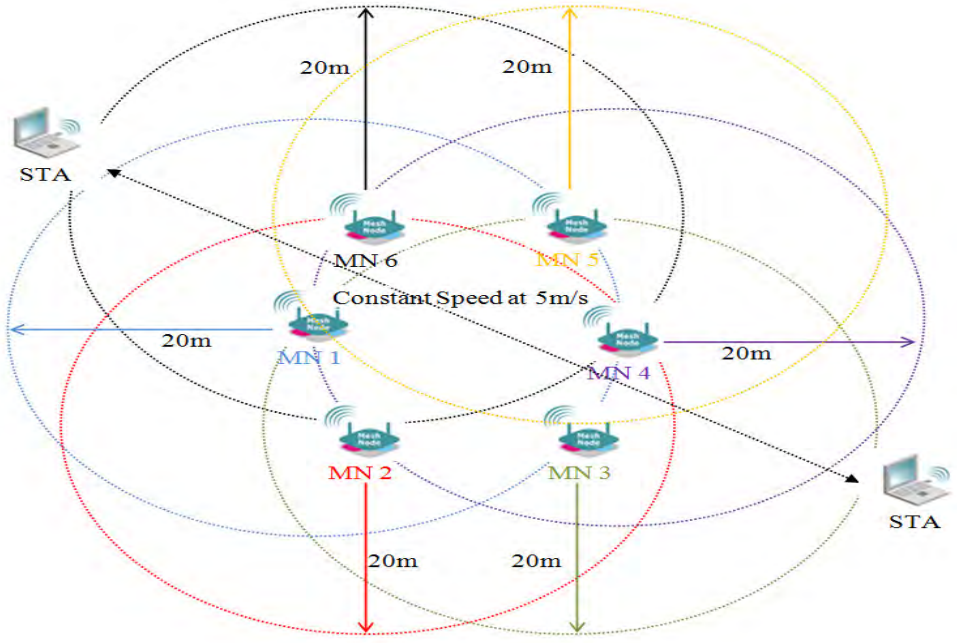


Figure 3.6: Six MNs Handoff Simulation Scenario

3.2.2 Experimental Work

There are a few limitations to using NS2 for simulating handoff. For example, wireless nodes assume that they are part of an ad-hoc networks and the PHY provided in NS2 only approximates the original 802.11 protocol. Furthermore, the PHY assumes a single channel and all the wireless nodes share this channel. Due to these NS2 limitations, it was only used to verify the operation of MeshScan. Experimental testing was necessary as an extension to the work of NS2 in order to more fully analyze the performance of MeshScan and to demonstrate its feasibility.

3.2.2.1 Testbed Detail

The experimental testbed included up to six MNs and five STAs. All MNs and all STAs had fixed positions to allow for repeatable experimental testing. All MNs and STAs had Ethernet connections for control proposes.

The MNs which were used were a part of CNRI mesh test which will be described in detail later in this chapter. All MNs had one physical WLAN NIC attached to them and madwifi driver version 0.94 was used as the driver which was configured to allow the MNs to have two virtual interfaces: *ath0* adhoc mode which allowed the MNs to communicate with each other and *ath1* in master mode which allowed MNs to create a service that looks like a traditional AP. Both virtual interfaces operated in 802.11a, channel 60 and open system key authentication was used for *ath1*. All adhoc virtual interfaces had the same essid (set to *mesh_handoff*) and all master virtual interfaces had the same essid (set to *ap_handoff*). All MNs acted individually as a BSS and collectively as an ESS, as shown in Figure 3.7.

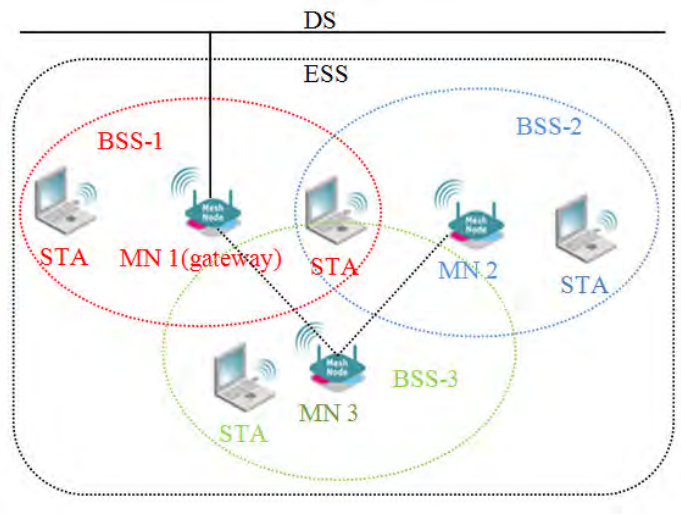


Figure 3.7: BSS and ESS Formations

Figure 3.8 illustrates the general experimental setup. The STAs were running Fedora 10 with a 2.6.27.24-170.2.68 kernel on Pentium PC platforms (Pentium(R) Dual-Core CPU E5200 2.5GHz, 1GB RAM) with an Atheros AR5212-based wireless interface on STAs 2-5. STA1 was configured as a WMN controller and used Secure Shell network protocol (SSH) to control all the other STAs and MNs. An automated perl script ran on the WMN controller to force client STA handoff among the MNs. On STA2, the WLAN NIC was configured to run in monitor mode which operated in 802.11a using channel 60 for monitoring the transmitted frame on the medium using the Wireshark open-source packet analyzer. STA3 was configured as a client STA where the NIC was configured to run in the STA mode and operated in 802.11a mode on channel 60 where open system key authentication was used. The location of the client STA was covered by all MNs individually to allow the STA to handoff among each of the MNs. STA4 and STA5, were configured to run in adhoc mode and operated in 802.11a on channel 60 where STA4 was used as traffic generator and STA5 was used as traffic sink (i.e. the receiver for the traffic

from STA4). The DITG network traffic generator tool was used on both STA4 and STA5. Table 3.3 outlines the equipment used.

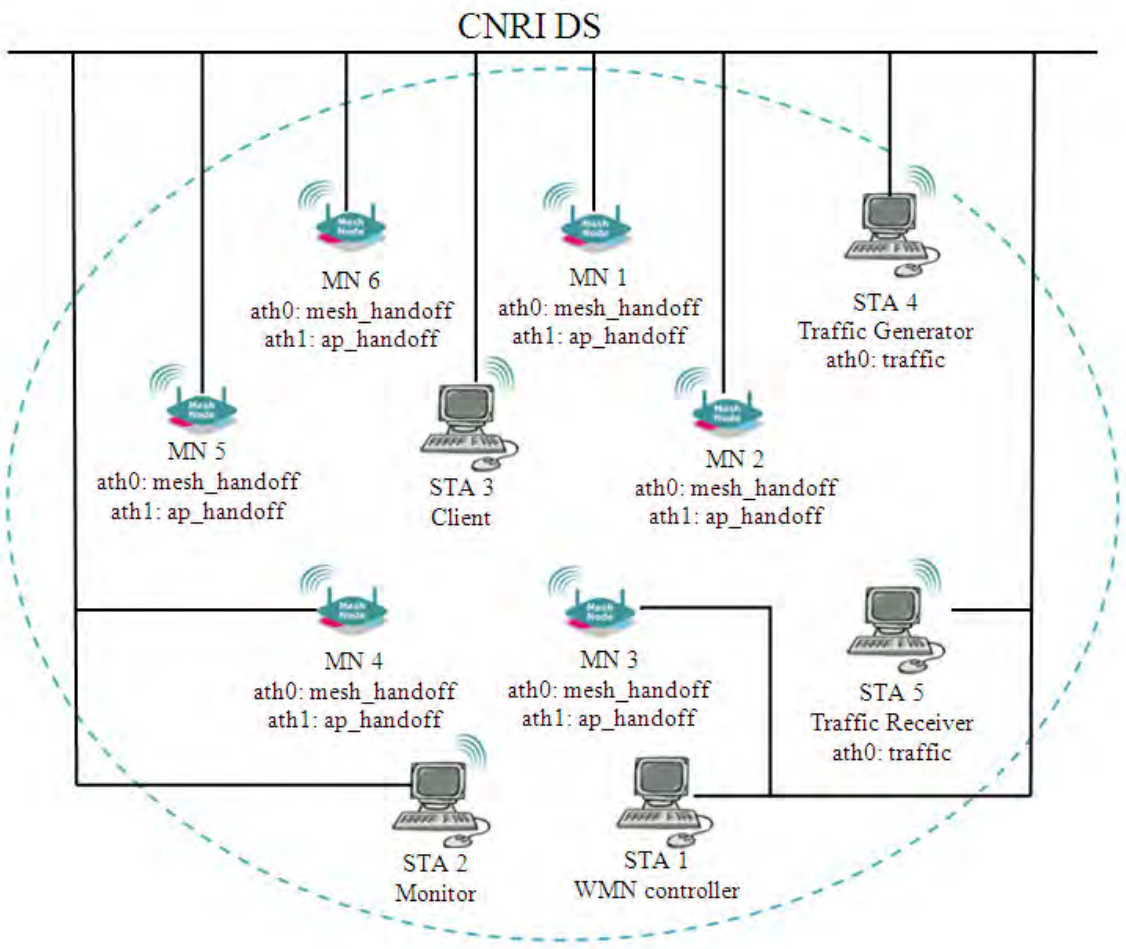


Figure 3.8: General Experimental Testbed Setup

Table 3.3: Testbed Equipment

Name	Model	OS	Radio	Eth IP Address	Mac Address
MN1	**Soekris net 4521**	**Pebble**	**Netgear WAG511**	**147.252.67.95**	**06:14:6c:5a:b4:09**
MN2	**Soekris net 4521**	**Pebble**	**Netgear WAG511**	**147.252.67.96**	**06:14:6c:5c:a3:a0**
MN3	**Soekris net 4521**	**Pebble**	**Netgear WAG511**	**147.252.67.210**	**06:14:6c:09:dd:0e**
MN4	**Soekris net 4521**	**Pebble**	**Netgear WAG511**	**147.252.67.211**	**06:09:5b:d2:ee:b3**
MN5	**Soekris net 4521**	**Pebble**	**Netgear WAG511**	**147.252.67.212**	**06:14:2f:af:52:79**
MN6	**Soekris net 4521**	**Pebble**	**Netgear WAG511**	**147.252.67.213**	**06:14:6c:09:dc:fa**
STA1/WMN controller	**Dell Optiplex 360**	**Fedora 10**	**NA**	**147.252.67.201**	**NA**

STA2/Monitor	**Dell Optiplex 360**	**Fedora 10**	**Netgear WAG511**	**147.252.67.203**	**06:14:6c:34:3f:aa**
STA3/Client	**Dell Optiplex 360**	**Fedora 10**	**Netgear WAG511**	**147.252.67.209**	**06:14:6c:34:3f:aa**
STA4 /Traffic generator	**Dell Optiplex 360**	**Fedora 10**	**Netgear WAG511**	**147.252.67.219**	**06:14:6c:aa:e0:15**
STA5 /Traffic receiver	**Dell Optiplex 360**	**Fedora 10**	**Netgear WAG511**	**147.252.67.220**	**06:14:6c:aa:e0:a1**

3.2.2.2 Experimental Test Scenarios

The experimental tests were divided into two groups corresponding to passive handoff and active handoff. Two automated perl scripts were developed which ran on the WMN controller and forced client STA handoff among MNs under passive handoff and active handoff. The SSH protocol was used to establish a secure connection to either MNs or client STA to control them remotely and to kill the connection after the session finished.

Passive handoff is triggered when the client STA receives a *disassociation management frame* which was sent by the current associated MN. Therefore the perl script was used to switch the MN's interface *ath1* on and off in order to force client STA to handoff among the MNs. The procedure of the automated perl script is as follows: Firstly, the WMN controller would check for the logged MN which was previously associated with STA. If there was a logged MN, the WMN controller would switch on that MN's *ath1* interface. Secondly, the WMN controller would check the MN which the client STA is currently associated with. If the client STA is not associated with any MN, no further action would be taken. Finally, if the client STA is associated with a MN, the WMN controller would connect to the particular MN which was currently associated with client STA and switch off its *ath1*. Before the *ath1* is switched off, the *disassociation management frame* would

be sent to the client STA to trigger the handoff process. Figure 3.9 outlines the procedure of the automated script for passive handoff.

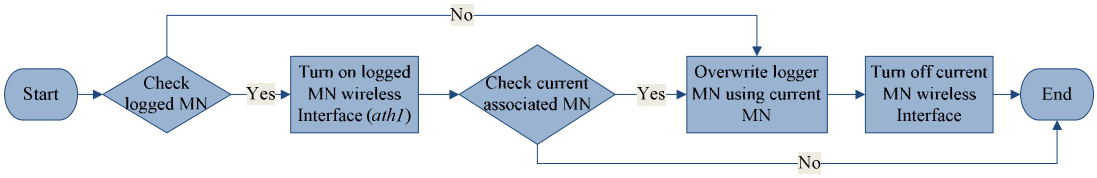


Figure 3.9: Procedure of Automated Script for Passive Handoff

Active handoff would be triggered when the current associated MN's *rssi* value was below the *rssi threshold*. Therefore the automated perl script was controlling (i.e. tuning) the transmission power of the MN's interface *ath1* which was associated with the client STA in order to force it to handoff among the MNs. The procedure of the automated perl script is as follows: Firstly, the WMN controller would check for the logged MN which was previously associated with the STA. If there was a logged MN, the WMN controller would increase the transmission power to the default value (18dBm in madwifi driver) on the logged MN's *ath1* interface. Secondly, the WMN controller would check for the MN which the client STA is currently associated with. If the client STA was not associated with any MN, not further action would be taken. Finally, if the client STA was associated with a MN, the WMN controller would connect to the particular MN which was currently associated with the client STA and reduce the transmission power of *ath1* in steps of 3 dBm until it reached 0dBm. This caused the *rssi* value of this particular MN to drop and eventually an active handoff would be trigged. Figure 3.10 outlines the procedure of automated script for active handoff.

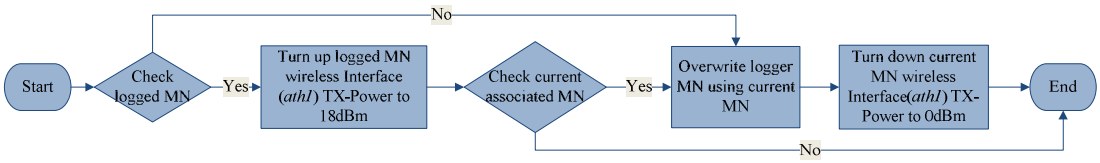


Figure 3.10: Procedure of Automated Script for Active Handoff

Different numbers of MNs were used in each experimental test. Also different network conditions (i.e. different network traffic loads) were used in each experiment. Table 3.4 outlines these experimental test scenarios in detail. Both groups used the same test scenarios for both passive and active handoff, therefore a total of 40 experiments were conducted. The experimental data was generated and stored by the client STA. The raw data includes all management frames that the STA transmitted and received during the handoff process with a timestamp.

Table 3.4: Test Scenarios Details

Test No	No of MNs	BG Traffic	Test No	No of MNs	BG Traffic
1	3	0	11	5	0
2	3	10	12	5	10
3	3	15	13	5	15
4	3	20	14	5	20
5	3	25	15	5	25
6	4	0	16	6	0
7	4	10	17	6	10
8	4	15	18	6	15
9	4	20	19	6	20
10	4	25	20	6	25

3.3 Software & Mesh Testbed

In this section, all the software applications that were used in this project are described. The details of the WLAN mesh test bed that was used for the experimental study are also described.

3.3.1 Network Simulator 2 (NS2)

NS2 [74] is a widely used software tool to simulate the behaviour of wired and wireless networks at a packet-level. It is an object-oriented, discrete event driven network simulator developed at UC Berkeley. NS2 includes wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. NS2 provides good flexibility to allow user easily modify NS2 to meet their needs. The version used was NS2-2.33.

3.3.2 Madwifi Driver

Madwifi driver [75] is open source Linux kernel driver for wireless LAN chipsets from Atheros [76]. The driver itself is open source but relies on a proprietary Hardware Abstraction Layer (HAL) [77]. The driver provides great flexibility for further development of WLAN management functions. Available interface modes include:

- STA mode allows a computer with a wireless network interface card (NIC) to operate in infrastructure mode and to connect with wireless AP.
- AP mode allows a computer with a wireless NIC to operate in infrastructure mode and to function as an AP.
- Monitor mode allows a computer with a NIC to monitor all traffic received from the wireless network.

- Ad-hoc mode allows a computer with a wireless NIC to operate in what the standard refers to as an independent basic service set (IBSS) network configuration

3.3.3 D-ITG Tool

D-ITG [78] (Distributed Internet Traffic Generator) is a network performance tool capable of producing traffic at a packet level which can be used to apply different network loads from 10Mbps to 30Mbps in order to subject the MeshScan to different network load conditions.

3.3.4 Wireshark

Wireshark [79] is a free packet analyzer application. It is used for network troubleshooting, analysis, software and communications protocol development. In this project, Wireshark was used to monitor frames that were transmitted from both the MNs and the client STA during handoff process in order to debug the MeshScan implementation in Linux.

3.3.5 Mesh Testbed

All experiments have been carried out using the CNRI wireless mesh testbed [73]. This testbed is a multi-purpose experimental networking platform which consists of 17 IEEE 802.11abg based mesh nodes, located around the Focus building at the Dublin Institute of Technology. Each MN is based upon a Soekris net 4521 platform as shown in Figure 3.11 and a NETGEAR WAG511 wireless adapter card. Each MN runs under Pebble Linux and uses the madwifi version 0.9.4 as the wireless network interface driver.

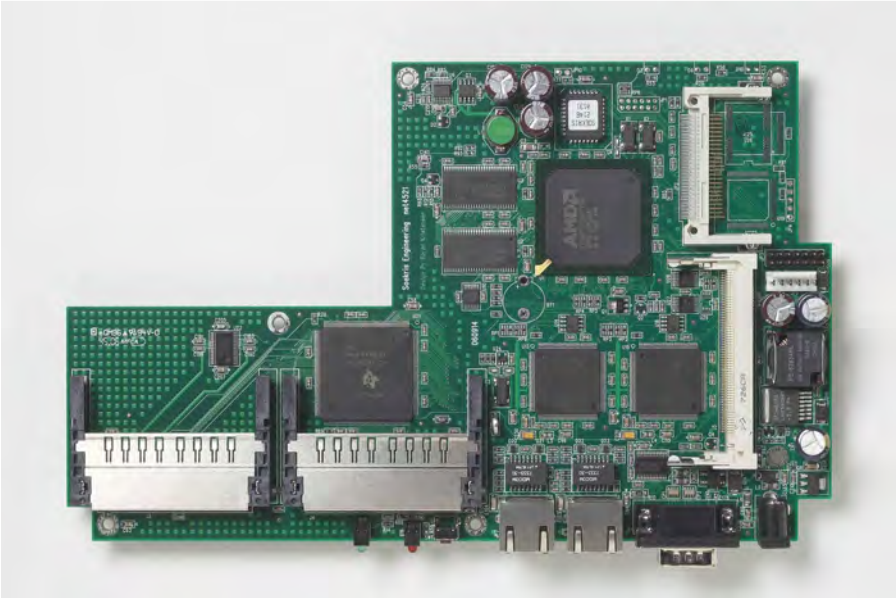


Figure 3.11: Soekris net4521 Platform

3.4 Chapter Summary

This chapter has outlined the different phases of this study including handoff analysis, the operation of the MeshScan scheme and the validation of MeshScan. A detailed description of validation of MeshScan was given including the simulation topologies, simulation scenarios, experimental testbed and experimental scenarios used. The chapter ended with an outline of the various software tools used in the study. The next chapter will outline the implement of MeshScan in both NS2 and Madwifi.

4. MeshScan Implementation Details

As discussed in the previous chapter, a client-side handoff scheme have been developed, called MeshScan, to provide for fast handoff which is less than 50 ms. MeshScan has been implemented on both NS2 and madwifi driver in this project where NS2 was used in order to demonstrate the feasibility of MeshScan and the madwifi driver was used to develop an experimental prototype of MeshScan in order to analyze its performance through experiments. Modifications were required to implement the MeshScan scheme in both NS2 and the madwifi driver. The basis of the MeshScan scheme is to decrease the total handoff duration by reducing the latency of the discovery phase. The modifications were made on NS2 and madwifi in the MAC layer where the discovery process is defined and implemented. This chapter describes the modifications made to both NS2 and the madwifi driver.

4.1 NS2 Simulator

NS2 is a network simulator which is the result of an on-going open source project. NS2 provides great flexibility to researchers in that their ideas can be quickly implemented and simulated though NS2 instead of having to develop a new simulation tool. NS2 is an object oriented simulator, written in C++, with an OTcl interpreter as a front end. The reason why NS2 needs two languages is that C++ is fast to run but slower to modify, making it suitable for detailed protocol implementation. OTcl runs much slower but can be modified very quickly and interactively, making it ideal for simulation configuration. The modifications were required in both C++ and OTcl sides for implementing MeshScan scheme in NS2.

The version of NS2 used in project was 2.33 which supports WLAN infrastructure mode simulation where Beacon frame, Scanning, Authentication and Association functions have been implemented. Handoff between nodes is also supported by NS2 which follows the IEEE 802.11 standard in three phases: discovery, authentication and association. The handoff detection is implemented to trigger the handoff in NS2. Figure 4.1 illustrates the client-side handoff process in NS2 [80].

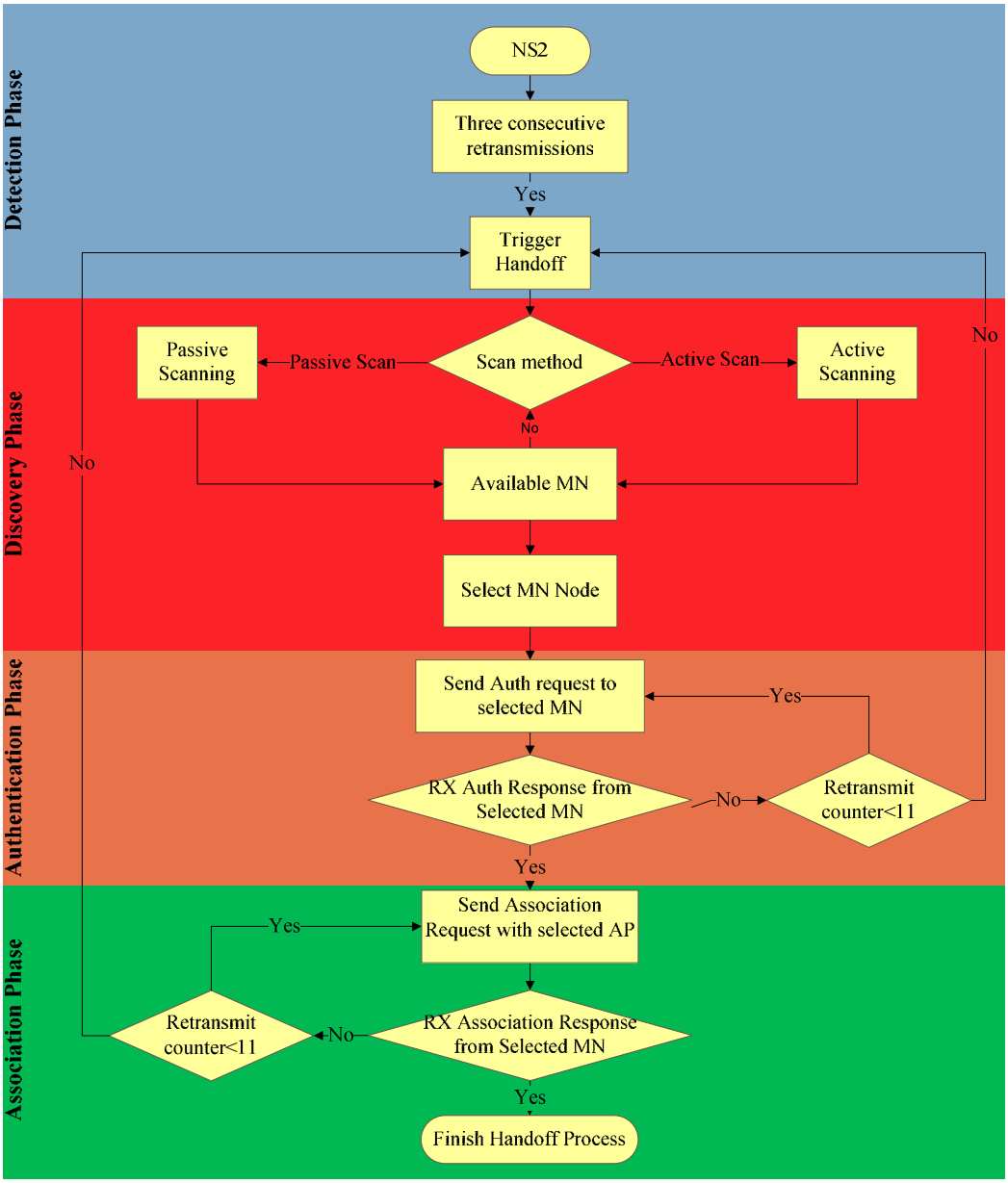


Figure 4.1: Client-Side Handoff Process in NS2

Handoff Detection – Three consecutive retransmissions indicate that a client node is moving out of range of an AP node. The other reasons for dropped packets such as collisions and fading are not yet supported by NS2.

Discovery – Once handoff is detected, either passive or active scanning is initiated to scan the all channels in order to find a MN which is available to the client node to associate with. NS2 only provides for a single channel wireless environment simulation and therefore NS2 is not suitable for simulating scan methods.

Authentication and Association – Once the client node selects the MN to associate with, the authentication and association procedures follow the IEEE 802.11 standard (as described in Section 2.5) to establish a connection between the client and MN. Four management frames are exchanged between the client STA and MN: authentication frame sent from the client STA, authentication frames sent from the MN, association request frame sent from the client STA, and association response frames sent from the MN.

According to the concept of MeshScan outlined in Section 3.2, the only modification required is within the discovery phase where MeshScan is used instead of either passive scanning or active scanning to provide for fast handoff.

Therefore a linked list and the Mesh Scan scheme are added to NS2. The linked list is used as a MN list to store MN information. A node identification number is used, instead of the MAC address, to identify a MN in NS2 The MN's information is stored in the Mesh list when a MN object is created. As described in chapter 3.2, an authentication request will be sent out to each MN in the MN list as a unicast scan. If an authentication response from a

MN is received, MeshScan stops transmitting unicast authentication request frames and MeshScan finishes. Figure 4.2 outlines the MeshScan procedures.

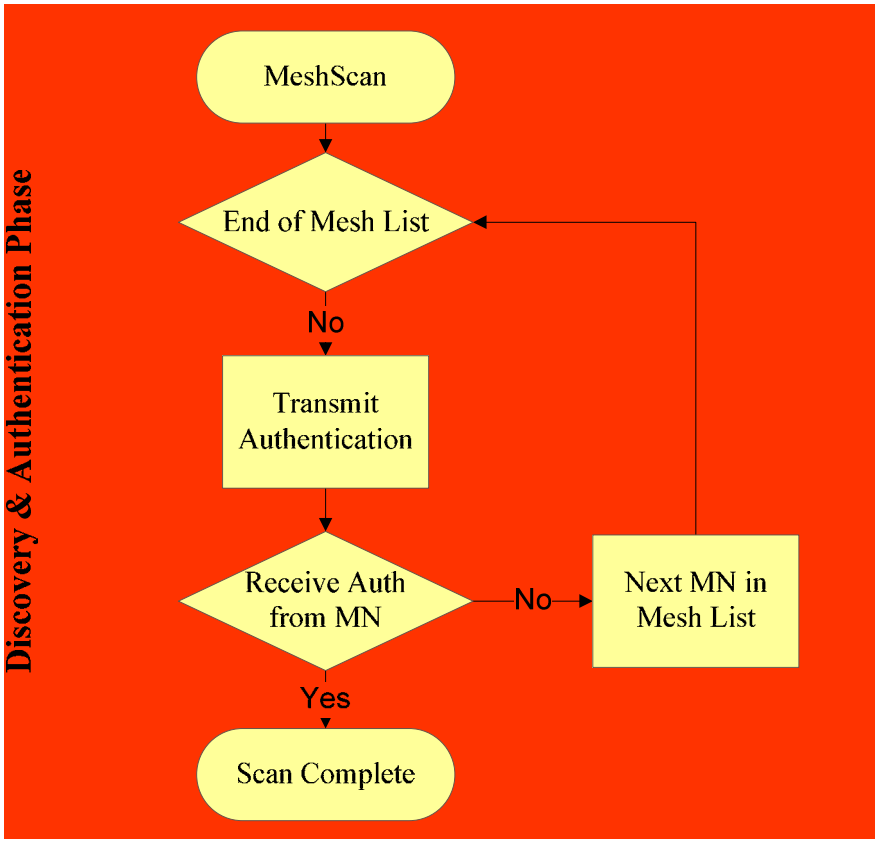


Figure 4.2: MeshScan Procedure

After the authentication request frames have been sent to all MNs on the MN list (end of the list, the client node will perform either passive or active scanning immediately in order to prevent introducing further latency. If an authentication response frame is received while passive or active scanning is operating, the client node stops scanning and associates with the MN which sends the received authentication response frame. If multiple authentication response frames are received, the client node only responses to the first authentication and ignores the rest of authentication response frames. Figure 4.3 outlines the handoff process using the MeshScan function.

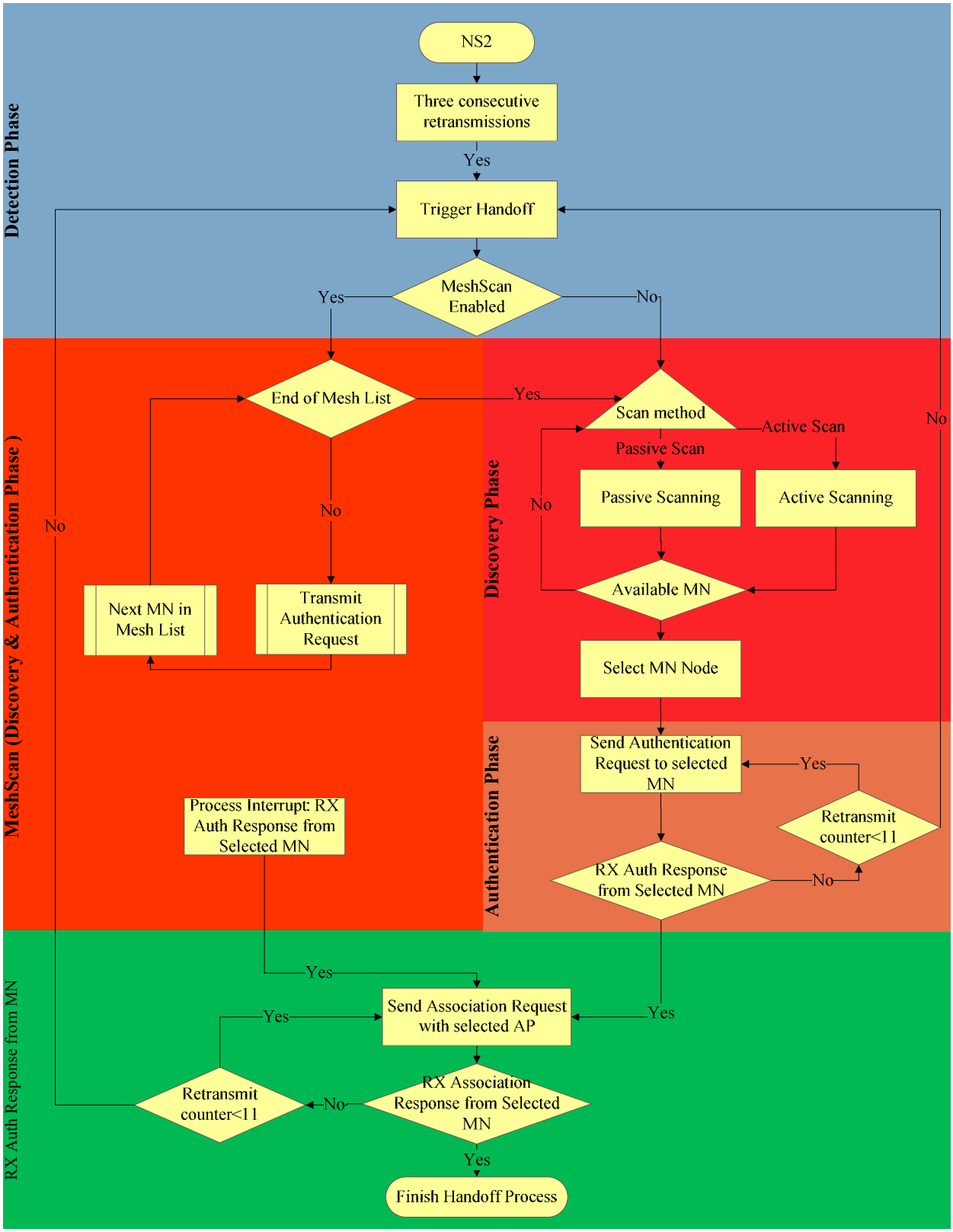


Figure 4.3: Handoff Process with MeshScan Function

As shown in Figure 4.3, there is no modification required to the IEEE 802.11 standard handoff phases. However, a new MeshScan phase is added in parallel with the discovery phase and authentication phase and a MeshScan trigger is added into the detection phase. The handoff process checks whether MeshScan is enabled or not when a handoff is triggered. If MeshScan is enabled, MeshScan will be used instead of the discovery and authentication phases. If MeshScan is not enabled, the IEEE 802.11 standard handoff process will take place as normal. Additionally, MeshScan phase combines two functionalities in the discovery and authentication phase (scan and authentication) in order to improve handoff latency dramatically. The IEEE 802.11 standard handoff process will be used after MeshScan fails to receive any authentication response from MNs which are stored in the MN List.

4.2 Madwifi Driver

As discussed in Section 3.3.2, the madwifi driver is an open source Linux kernel driver for WLAN chipsets from Atheros. However, the madwifi driver depends on a proprietary Hardware Abstraction Layer (HAL) which is available in binary form only and acts as an API to present the hardware to the driver in a uniform fashion. Madwifi is one of the most advanced drivers for WLAN devices for Linux today with a wide set of features. Using the madwifi driver, multiple virtual interfaces can be created on one physical network card. Each virtual interface can work in different modes, namely AP, STA, adhoc, or Monitor.

As a Linux kernel driver, madwifi is written in C and uses *ioctl* as the user to kernel interface to get and set the madwifi parameters such as *rssi threshold, transmit power, MAC retry* etc. Madwifi comes with a dynamic debug tool called *80211debug* for supporting madwifi development.

The version of madwifi driver used in project was 0.9.4 which is the most current stable release in 2009. As described in Section 3.2, MeshScan is designed to work on client STAs, therefore the modification of madwifi is activated when a virtual interface is configured in STA mode. In order to carry out the modifications to the madwifi driver, the behaviour of the STA operations according to madwifi driver was studied. Figure 4.4 outlines the state diagram of STA mode in the original madwifi driver [81].

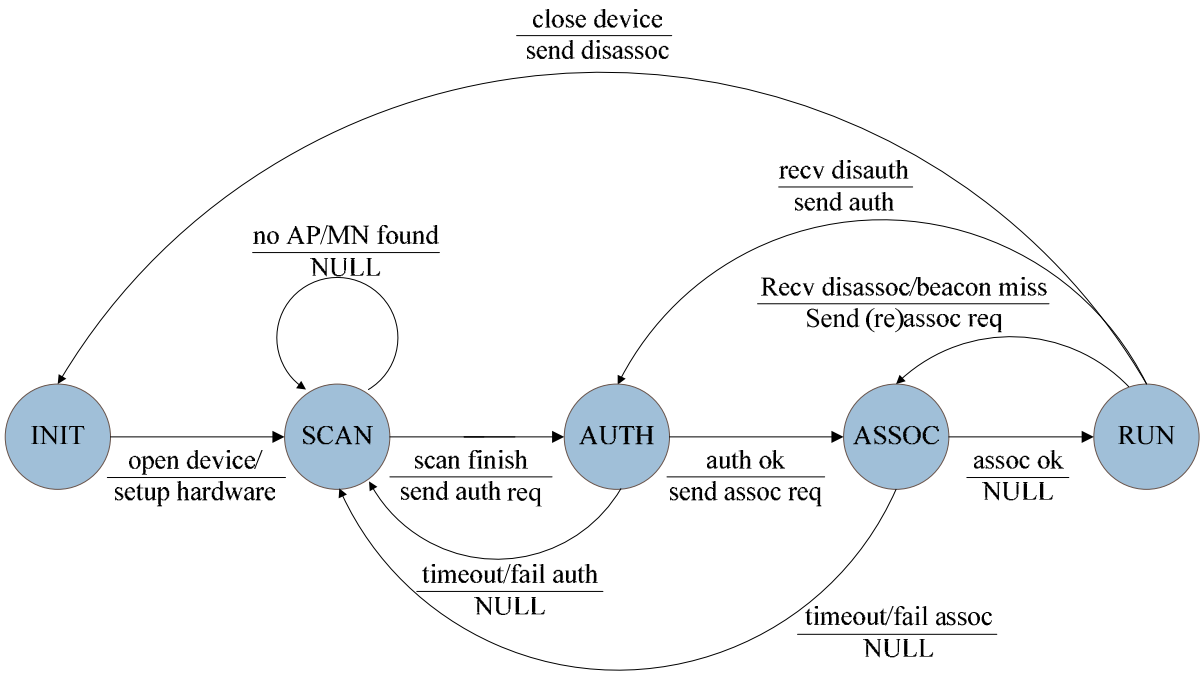


Figure 4.4: Madwifi Driver - State Diagram of STA Mode

When the driver is loaded, it searches the physical network card and then sets up the madwifi device. The driver also automatically creates a virtual network interface operating in the specified mode. By default, a STA mode virtual interface will be created. The initial state of the virtual interface is INIT. In the initial state, the interface parameters will be

configured, such as *scan method*, *transmit power*, *rssi threshold* etc. While in the INIT state, the hardware does not receive or transmit packets.

When the virtual interface is switched on (for example, by using the command *ifconfig ath0 up*), the driver configures the hardware and enters the SCAN state. In the SCAN state, the STA scans all the supported channels using either passive scanning or active scanning. After the scan is complete, the STA selects one AP/MN that has the desired *essid* and the highest *rssi*. If no AP/MN with a matching *essid* is found, the STA restarts a new scan. If an AP/MN is selected, the STA configures the parameters required to communicate with the AP/MN, and then enters the AUTH state.

On entering the AUTH state, the STA starts an authentication procedure by sending an authentication request frame to the selected AP/MN. The authentication procedure includes a sequence of messages exchanged between the STA and AP which depends on whether Open System authentication or Shared Key authentication is used. If the authentication succeeds, the STA enters the ASSOC state. Madwifi defines two variables to prevent transmission failure according to IEEE 802.11 standard: *IEEE80211_TRANS_WAIT* defines the timeout time for a transmission to be considered as failure and leads to a retransmission. *ATH_TXMAXTRY* defines the max retransmission threshold for the number of times that a packet can be retransmitted. In madwifi, the default value for *IEEE80211_TRANS_WAIT* is 5 seconds and the default value for *ATH_TXMAXTRY* is 11 attempts.

On entering the ASSOC state, the STA sends an association request frame to the AP/MN and waits for an association response to establish connection. If the STA receives a successful association response, it goes into the RUN state. If the association fails (e.g. an

error response or rate negotiation failure), or if the STA does not receive any association response after reaches *ATH_TXMAXTRY* threshold, the STA goes into the SCAN state.

In the RUN state, the STA can exchange data packets with the AP/MN. The STA also listens to management messages. If the STA receives a disassociation frame from the current associated AP/MN, it sends an association request frame and goes into the ASSOC state. If the STA receives a disauthentication frame, it sends an authentication request frame and goes into the AUTH state. In the RUN state, the STA maintains the connectivity to the AP/MN by listening to beacon frame. If 10 consecutive beacons are missed, the connection to the AP/MN is considered broken. The STA sends a re-association request to the AP/MN and enters the ASSOC state to try to reassociate with the AP/MN.

From the above procedure the handoff procedure can be identified for a STA. The handoff procedure starts when the STA is in the RUN state and fails to receive 10 consecutive beacon frames or receives a disassociation frame. The STA sends a reassociation request to the old AP/MN and enters the ASSOC state. In the absence of a reply from the old AP/MN, the STA enters the SCAN state to search for any new AP/MN. Figure 4.5 outlines the handoff procedure in the original madwifi driver. A reassociation phase is introduced by madwifi to try to establish a connection when handoff is required as an extension of detection phase.

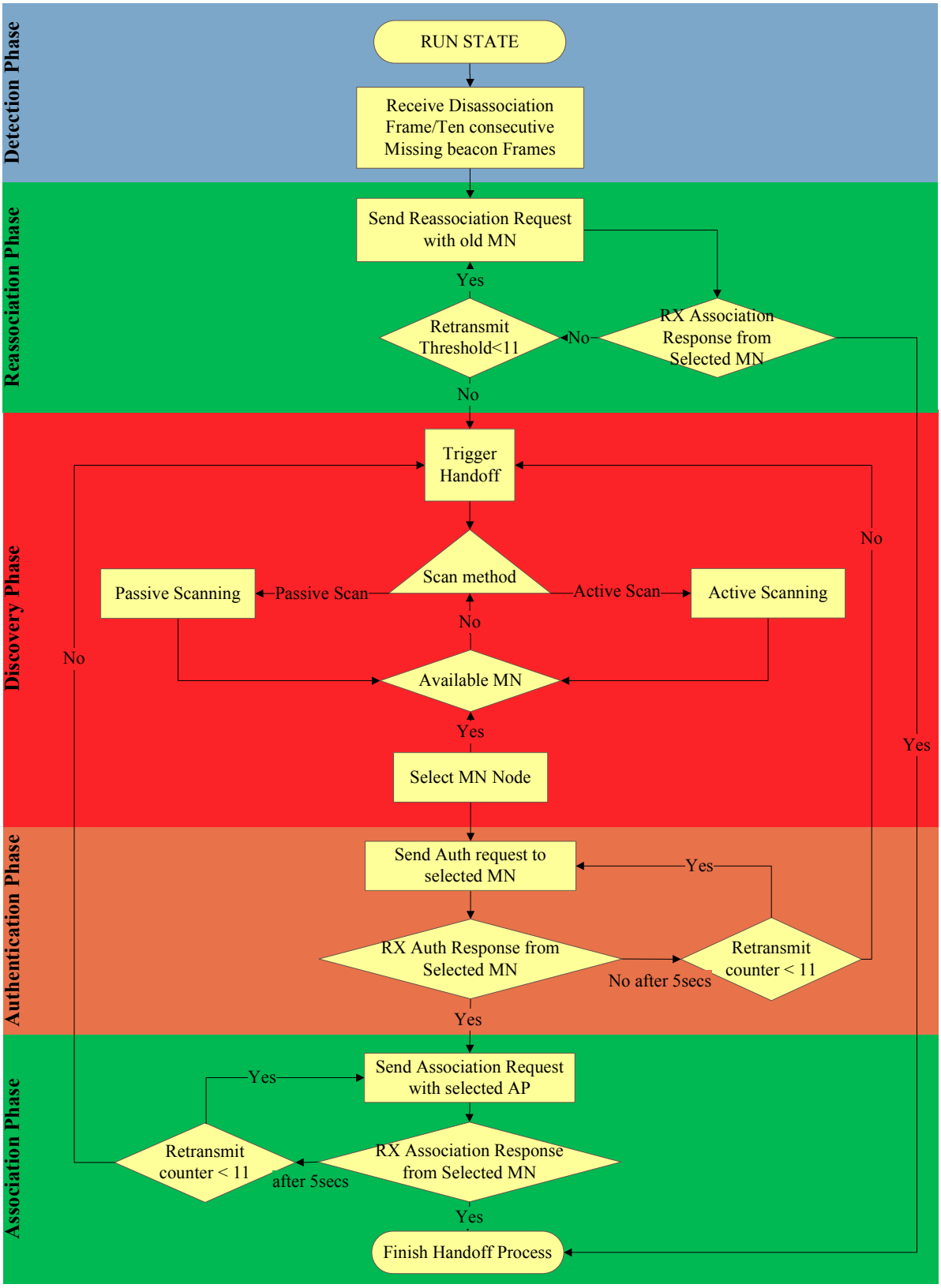


Figure 4.5: Handoff Procedure in Original Madwifi Driver

The modification to the madwifi driver is divided into two main parts: changes to the kernel driver and a userspace interface to control the handoff procedure. The madwifi driver changes are the minimum required to support the new functionalities, system kernel log, SmartList, MeshScan state, userspace interface and MeshScan handoff procedure.

System kernel log is the output file from the madwifi driver. The kernel log records all handoff related management frame (e.g. disassociation frame, authentication frames and association frames) exchange between the STA and AP/MN. The timestamp is also captured for each management frame when madwifi handles the management frame. System kernel log provides reliable and precise experimental data in order to obtain accurate results for all experiments carried using madwifi.

The SmartList is a linked list as shown in Figure 4.6 where the MN information stores (MAC address and *rssi* value) and manages the MNs. (In this work, the SmartList preloaded onto client side in order to perform MeshScan because of this work aim to study to feasibility and performance of MeshScan scan technique.) The list is ordered where a MN's position on the list depends on its *rssi* value. The MN with the highest *rssi* value will be put at the top of the list in order to provide fast handoff to the best available MN. The *rssi* value is calculated on the captured beacon frame's *rssi* value dynamically. Because the *rssi* value is not constant, due to effects like fast fading and mobility of the environment. [17], an Exponential Moving Average (EMA) filter is used to obtain an average *rssi* value in order to mitigate the effects of interference and channel fading etc. Here *e_rssi* is the average of *rssi* over a time period of *T*, with the smoothing factor α set to 0.3 in this scheme, as shown in Equation 4.1.

$$e\_rssi_T = \alpha \times rssi_T + (1-\alpha) e\_rssi_{T-1} \quad (4.1)$$

Figure 4.6: SmartList

The madwifi MAC filter function was modified and was used as a userspace interface. It provides for flexibility and simplicity when adding or removing MNs from the SmartList which is central to the MeshScan scheme.

When handoff is required, the mobile client performs a unicast scan by transmitting Authentication Request frames to each of the MNs on the list to discover the next MN for handoff. In order to perform the MeshScan quickly and effectively both the retransmission threshold *ATH_TXMAXTRY* and the transmission waiting threshold *IEEE80211_TRANS_WAIT* are set to their minimum values where *ATH_TXMAXTRY* is set to 1 and *IEEE80211_TRANS_WAIT* is set to 1 ms. Figure 4.7 outlines the state diagram of STA mode in the modified madwifi driver containing the MeshScan functionality.

As two events can trigger handoff – receiving a disassociation frame or the captured frames has a low *rssi* - handoff can be divided into passive handoff and active handoff. In passive handoff, the STA does not have control over when handoff should be triggered but will be informed. (e.g. by receiving a disassociation frame or failing to detect 10 consecutive beacon frames). In active handoff, the STA does have control over when handoff should take place based on *rssi_handoff_threshold* which are new variables that are

added to support active handoff in madwifi driver. The active handoff is designed for triggering handoff before the connection become unavailable.

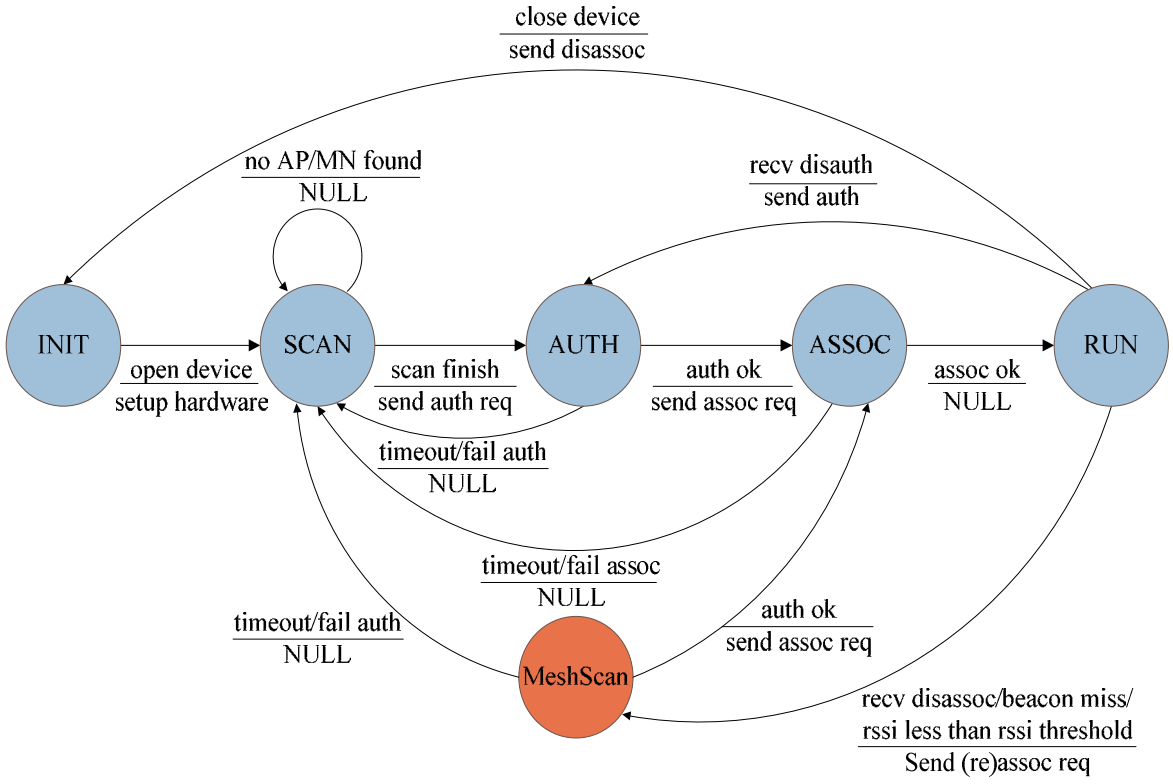


Figure 4.7: MeshScan Enabled Madwifi Driver: State Diagram of STA Mode

$$rssi\_handoff\_threshold = rssi\_factor \times rssi\_threshold \quad (4.2)$$

The *rssi_handoff_threshold* is calculated from *rssi_threshold* as shown in Equation 4.2. The *rssi_handoff_threshold* is the *rssi* threshold used to trigger handoff in active handoff in madwifi driver. The *rssi_threshold* variable is used to define a radio signal strength which

is not sufficiently strong to maintain the connection in the madwifi driver. The *rssi_factor* is used to define *rssi_handoff_threshold* from *rssi_threshold*. The reason for using *rssi_handoff_threshold* instead of *rssi_theshold* is because madwifi considers this connection is no longer available when the *rssi* value falls below *rssi_threshold* which is set to 9 by default in the madwifi driver. In general, an *rssi* of 10 or less represents a weak signal. An minimum *rssi* of 20 is considered acceptable for ensuring a reliable connection. An *rssi* of 40 or more is considered a strong signal. The *rssi_threshold* can be changed easily through the *ioctl* interface which is provided by the madwifi driver.

Figure 4.8 outlines the handoff procedure is the modified madwifi driver with the MeshScan scheme. The handoff procedure is performed with the following steps. When handoff is performed the transmit threshold and transmission waiting time are set to their minimum (as described earlier in this chapter) before it transmits an *Authentication Request* to each of the MNs on the SmartList and set back to default value afterwards. When the first *Authentication Response* is received, the STA stops transmitting *Authentication Request* frames to the rest of the MNs on the SmartList and enters the association phase to complete the handoff process. In the case where no Authentication Response is received after all Authentication Requests have been transmitted to the MNs on the SmartList, the STA will reassociate with the old AP/MN if handoff is triggered by active handoff or the STA will perform either passive or active scanning to try to discover if any other wireless networks are available.

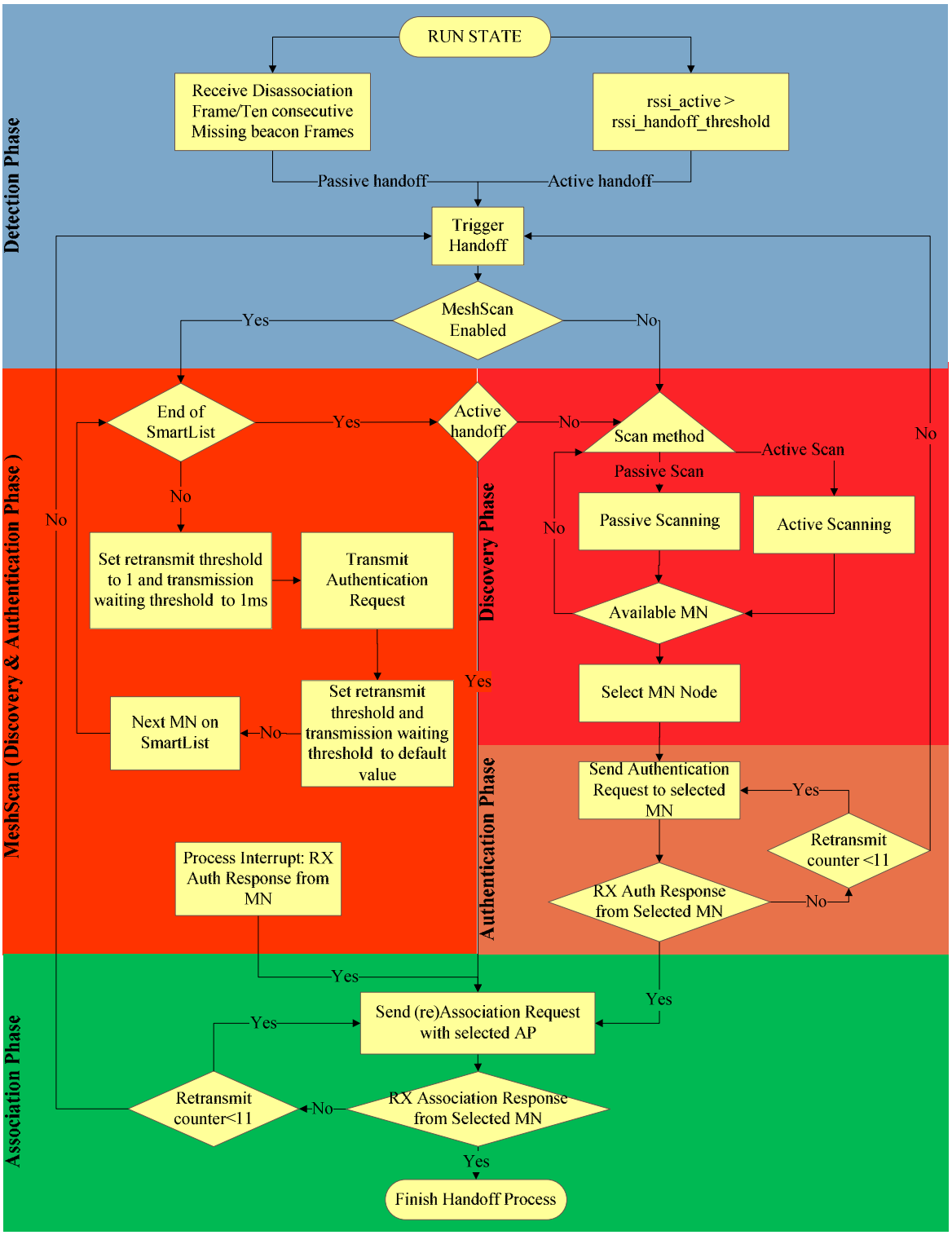


Figure 4.8: Handoff Procedure in modified Madwifi Driver with MeshScan Scheme

4.3 Chapter Summary

In this chapter, the details of the implementation of MeshScan was given for both the NS2 simulator version 2.33 and Madwifi driver version 0.94. The objective of carrying out the modification to NS2 is to verify the feasibility of the MeshScan technique and to compare its performance with the traditional scanning techniques (passive scanning and active scanning) defined in the IEEE 802.11 standard. The objective of the modification to the Madwifi driver is to develop a prototype in order to conduct an experimental performance test of MeshScan under different network conditions. The next chapter will present the results generated from the computer simulations and experiments.

5. Results and Analysis

In this section the experimental results are presented in detail and an explanation is provided. This section is divided into three sub sections corresponding to the work layout described in chapter 3. As discussed earlier in chapter 3, the first section presents the analysis of the IEEE 802.11 standard handoff scanning latency including a mathematical model for passive scanning and experiments for active scanning. The second section presents simulation results for MeshScan using NS2. The third section demonstrates the effectiveness of MeshScan through experiments for both passive and active scanning. Different network conditions (i.e. different network traffic loads) were used in each experiment. Two Tables in Appendices I listed detail network parameters used in both simulation and experiment.

5.1 IEEE 802.11 Handoff Analysis

This section describes studies that were performed to divide the total handoff latency into the discovery phase latency and the execution phase latency in order to determine which is primarily responsible for the unacceptable delay in the handoff process. The discovery phase latency is the time required to find the next MN to associate with. Two techniques were used: mathematic modelling and experiments for both passive and active scanning respectively. Execution phase latency is the time required to establish the connection to the chosen MN. An experimental approach was used to determine the latency introduced by the authentication and association frame exchanges.

5.1.1 Discovery Phase

5.1.1.1 Passive Scanning

As outlined in section 3.1.2.1.1 passive handoff latency can be calculated theoretically according to the *beacon_interval* where the passive scanning latency decreases as the *beacon_interval* reduces. A PHY layer PLCP preamble signals the beginning of a frame transmission and is used to prepare the wireless radios for communication. There are two preambles defined in the 802.11b/g standard, short and long. In 802.11a networks only the short preamble is allowed. Table 5.1 outlines the different values of *PLCP*, *DIFS*, *aver_backoff* according to preamble types in different 802.11 modes [82-83].

Table 5.1: PLCP, DIFS, and Aver_backoff Values in Different Preamble Type

	11a	11b Short Preamble	11b Long Preamble	11g Short Preamble	11g Long Preamble
PLCP	24 µs	96µs	192µs	96µs	192µs
DIFS	34 µs	50µs	50µs	50µs	50µs
Aver_backoff	67.5µs	310µs	310µs	310µs	160µs

The aim of this calculation is to measure the throughput loss for different values of *beacon_interval* in IEEE 802.11 networks. Figure 5.1 shows the impact on bandwidth when the *beacon_interval* varies from 100 ms to 10 ms.

The measure of the impact of transmitted beacon frames on the bandwidth capacity of APs is defined as the percentage of time consumed by the beacon frames. Figure 5.1 presents a graph of the impact for the different IEEE 802.11 modes. The results suggest that the beacon interval can be significantly reduced without causing significant bandwidth loss.

The results also show that lower beacon intervals in IEEE 802.11a networks have considerably less impact compared to IEEE 802.11b/g networks. At a *beacon interval* of 10 ms, the IEEE 802.11b/g standard suffers from an unacceptable 25% to 45% reduction in bandwidth. However, the IEEE 802.11a standard loses only 5% of its bandwidth.

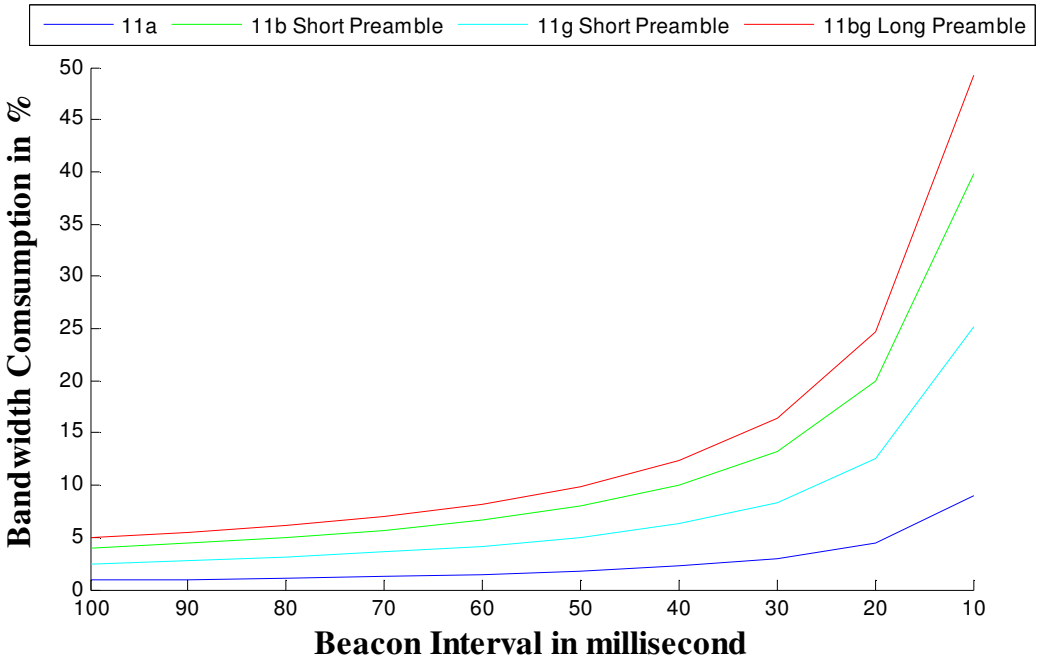


Figure 5.1: Bandwidth Consumption for Different Beacon Interval

In order to scan all available channels (13 available channels in Europe), passive scanning will take 130 ms when the *beacon_interval* is set to 10 ms in the IEEE 802.11a mode. In summary, the results from this section indicate that passive scanning is not suitable for applying fast handoff in both WLAN and WMN networks.

5.1.1.2 Active Scanning

As outlined in section 3.1.2.1.2, active scanning is a scan technique that can be used in the discovery phase. Figure 5.2 shows a PDF of the latency in milliseconds for active

scanning from experiments. The number of MNs was varied in each scenario from three up to six MNs. The dashed line is used as a reference to show the fast handoff target of 50ms.

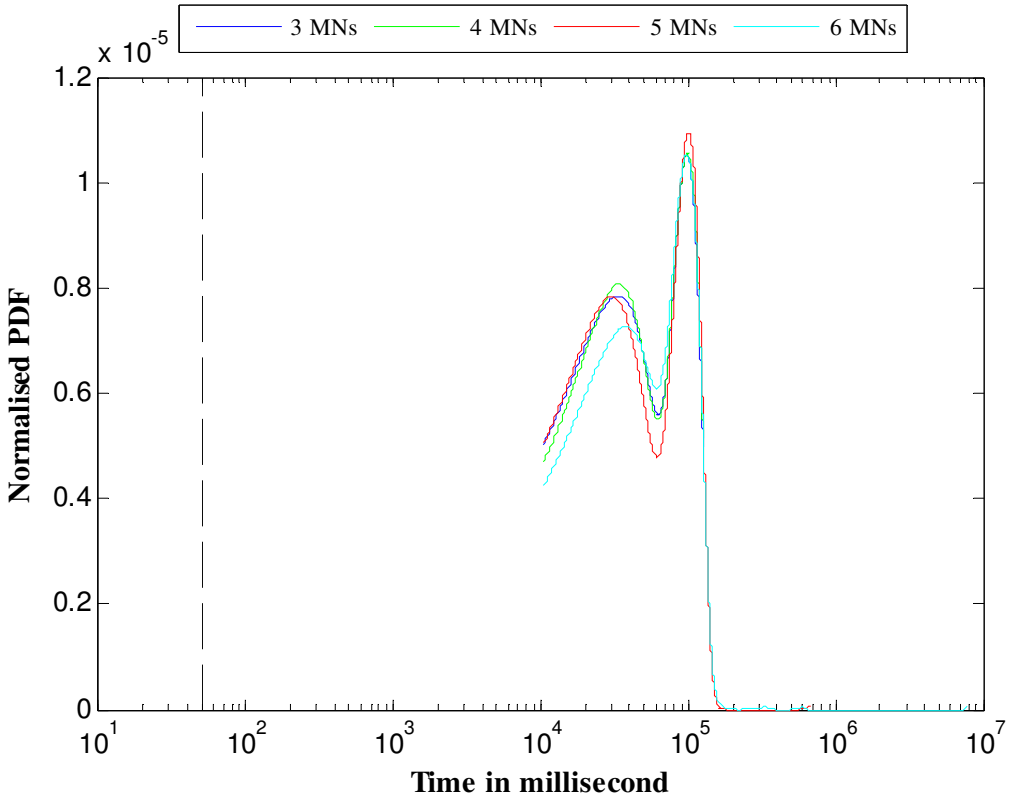


Figure 5.2: PDF of Active Scanning Latency from Experiment

Figure 5.2 shows that active scanning takes a considerable amount of time to scan through all available channels from 10sec to several minutes in all scenarios. Due to the increased contention it also shows a slight increase in the latency when the number of MNs was increased. A close inspection of the PDF indicates that all scenarios bear similar delay characteristics. (i.e. two peaks - the first peak at 15.8 seconds and the second peak at approximately 100 seconds). This characteristic indicates that most of the active scanning cases require the available channels to be scanned twice in order to complete handoff.

From the above plots, This conclusion can be made that neither passive scanning nor active scanning is suitable for implementing fast handoff as both techniques take several

seconds to scan all available channels. This latency cannot be tolerated for VoIP applications.

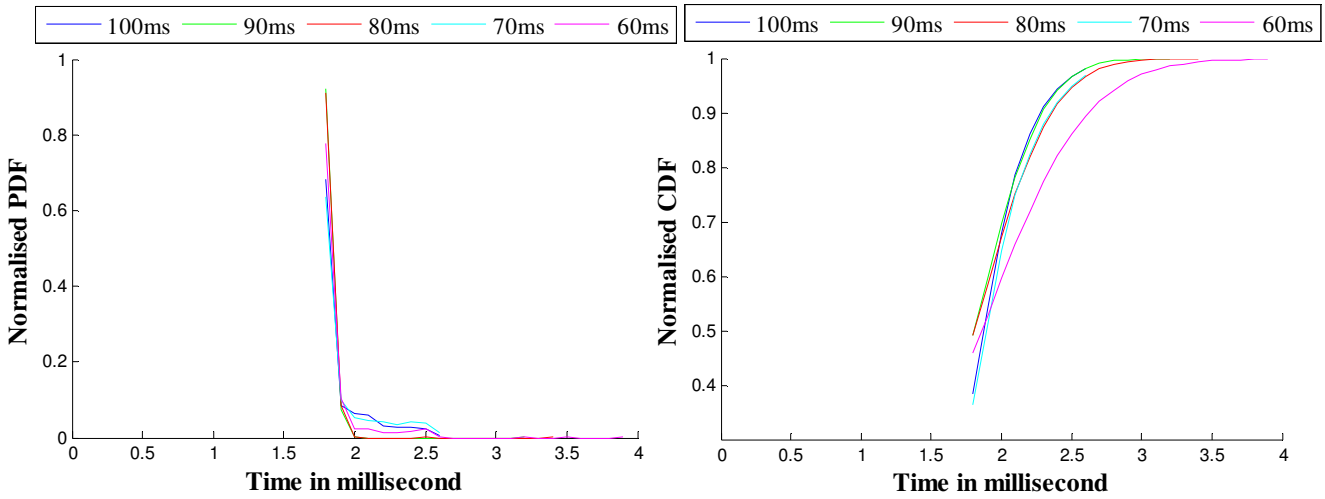
5.1.2 Execution Phase

As outlined in section 3.1.2.2, the execution phase includes an authentication phase and a association/re-association phase in the standard IEEE 802.11 handoff procedure. The execution phase latency begins with two frame exchanges, namely the authentication transaction and the (re)association transaction. Both simulation and experimental results were used to study the execution phase latency. The objective here is to study the execution phase latency in different network setups (i.e. by using different beacon intervals and with a different number of MNs).

5.1.2.1 Simulation

Figure 5.3 (a) shows a PDF plot of the simulation results for the execution phase latency in milliseconds for different beacon intervals which ranged from 60ms to 100ms. Figure 5.3 (b) shows the CDF plot of the same simulation results as Figure 5.3.

The PDF shows that the majority of execution phase latency is concentrated below approximately 1.9ms. It also indicates that execution phases for the different beacon intervals bear similar latency characteristics. The CDF shows that the probability of the execution phase latency having a value less than 3 ms is high (i.e. well over 95% of the latencies are below 3 ms). When decreasing the *beacon_interval*, the execution phase latency increases because of the extra beacon management frame overhead generated by the beacon frame. Again, the CDF clearly shows that approximately 99% of latencies (with exception of when beacon interval is 60 ms) are below 3 ms. In the case where the beacon interval is 60 ms, approximately 90% of latencies are below 3 ms.



(a) PDF of the Simulation Results

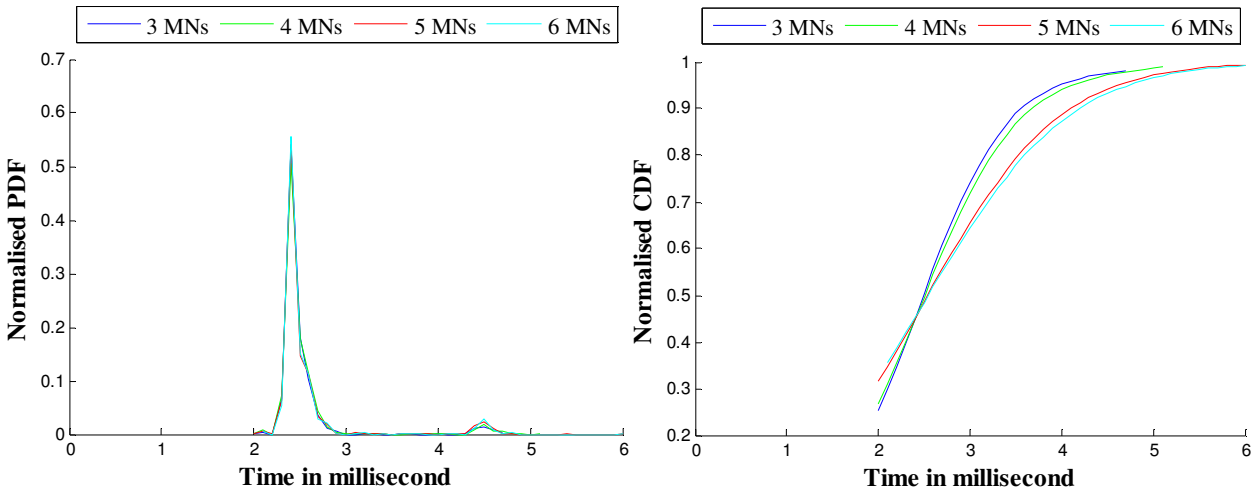
(b) CDF of the Simulation Results

Figure 5.3: Execution Phase Latency from Simulation

5.1.2.2 Experiment

Figure 5.4 (a) and Figure 5.4 (b) show the PDF and CDF respectively of the experimental results for the execution phase latency respectively. Different numbers of MNs were used in each scenario from three MNs to six MNs.

Figure 5.4 (a) shows that the majority of the execution latency is to be found between 2 ms and 4 ms. It also indicates that the execution phase latency for different numbers of MNs bear similar characteristics (i.e. approximately 57% of the latencies are centred around 2.5 ms). The CDF for the same experiments shows the probability of the execution phase latency being under 4 ms is high (i.e. well over 80% of latency values are below 4 ms). It is also clear that the execution phase suffers longer delays when the number of MNs increases due to the increasing contention on the medium.



(a): PDF of the Experimental Results

(b): CDF of the Experimental Results

Figure 5.4: Execution Phase Latency from Experiments

Comparing the simulation and experimental results shows that in both cases the execution phase latency is small, typically of the order of a few milliseconds.

5.1.3 Analysis Summary

The main goal of the work in this section was to investigate the delay associated with each step of the handoff process in order to gain a better understanding of what actually occurs during the handoff process. A secondary objective was to determine how much delay each phase of the handoff process introduces into the total latency. The results indicate that the discovery phase is responsible for over 99% of the total latency of the overall handoff process. The two standard scanning techniques (passive and active scanning) are not suitable for implementing fast handoff scheme because both of them require a scanning of all available channels to find a new MN to associate with. The delay introduced

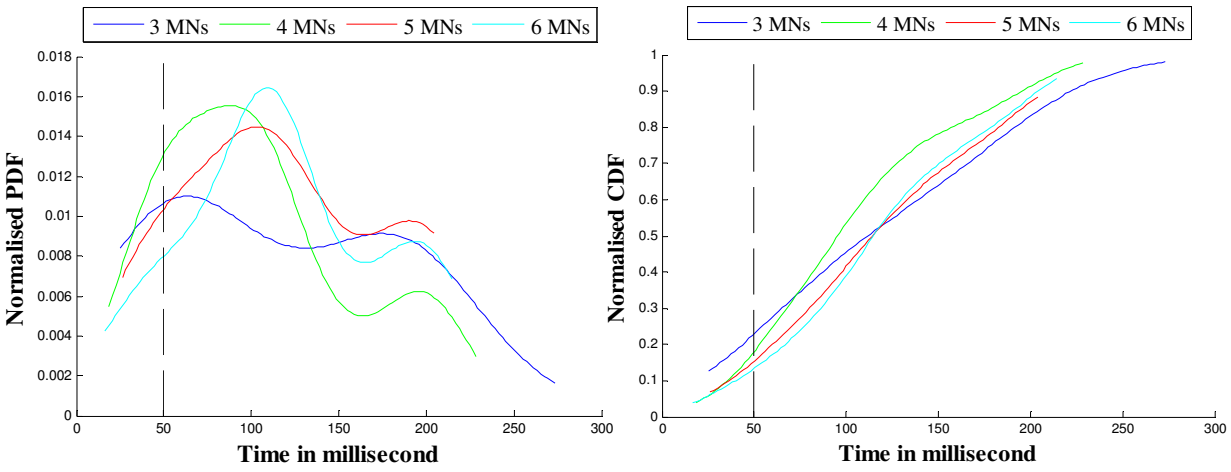
by the execution phase is of the order of a few milliseconds which is insignificant in terms of its contribution to the overall handoff latency compared to that of the scanning delay. Furthermore, the execution phase latency could not be further reduced unless the original IEEE 802.11 standard was to be changed. Also the recent IEEE 802.11r standard introduces a new Fast BSS Transition mechanism to enhance the execution phase for fast handoff. However, the standard does not address the question of when or to whom a STA should handoff to? Therefore a new scan technique needs to be developed to avoid having to scan all channels in order to realize fast handoff.

5.2 MeshScan Scheme Simulation

Following on the analysis of the handoff process, a fast handoff scheme called MeshScan was developed which is specifically focused on WMN applications. This section describes the computer simulations that were performed to verify the MeshScan Scheme. A set of different simulation scenarios was used to verify the feasibility of MeshScan and to compare its latency with that of the standard IEEE 802.11 handoff latency. The SmartList was preloaded to client manually before the simulation started in all cases.

5.2.1 Handoff Latency by Using Passive Scanning

Figure 5.5 (a) and Figure 5.5 (b) show the simulation results for handoff latency by using passive scanning during the discovery phase. A dashed reference line shows the fast handoff target of 50 ms.



(a) PDF of the Passive Scanning

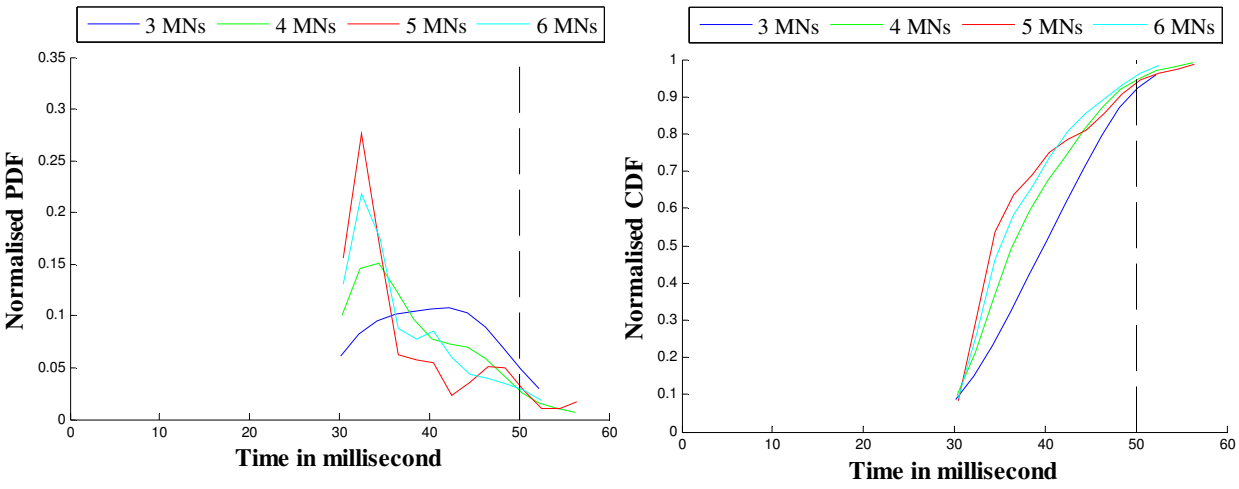
(b) CDF of the Passive Scanning

Figure 5.5: Simulation Passive Scanning Handoff Latency under Different Number of MNs

Figure 5.5 (a) shows that the handoff latencies under passive handoff appear to be widely distributed from approximately 25 ms to 275 ms when the mesh topology included three MNs. When the number of MNs is increased in the mesh topology (i.e. from four to six MNs), the spread of latency values are reduced from approximately 20 ms to 230 ms. This result indicates that passive scanning is not suitable for implementing a fast handoff scheme. A closer inspection of the PDF indicates that all scenarios bear similar delay characteristics apart from the three MNs scenario (i.e. the distribution contains two peaks where the: first peak latency is centred around 100 ms and the second peak centred around 200 ms). The two peaks in Figure 5.5 (a) indicate that the STA was moving between regions of low and high contention arising from the motion of the STA between the two MNs. Figure 5.6 (b) shows that approximately 80% of the handoff processes are completed within 200ms for all scenarios considered. The CDF also indicates that when the number of MNs is greater than four, the handoff process suffers longer latencies due the extra overhead (from the beacon frames) introduced by the MN.

5.2.2 Handoff Latency by Using Active Scanning

Figure 5.6 (a) and Figure 5.6 (b) show the simulation results for the handoff latency when using active scanning during the discovery phase. A dashed reference line shows the fast handoff target of 50 ms.



(a): PDF of the Active Scanning

(b): CDF of the Active Scanning

Figure 5.6: Simulation Active Scanning Handoff Latency under Different Number of MNs

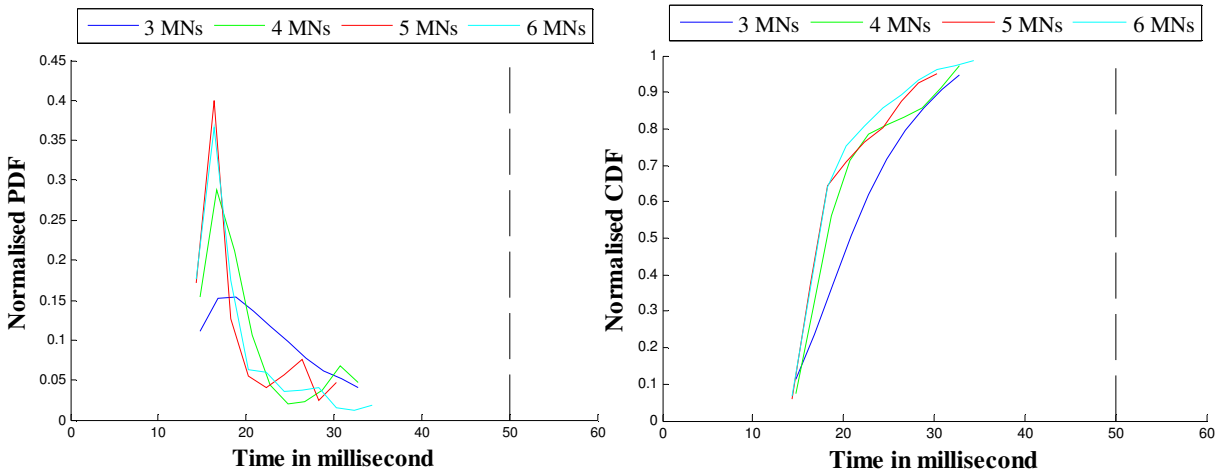
Figure 5.6 (a) shows that the handoff latencies under active scanning range from approximately 30 ms to 55 ms. All scenarios bear similar delay characteristics apart from the three MNs scenario (i.e. which exhibits a single peak centred around 33 ms.). Figure 5.6 (b) shows that approximately 90% of the handoff processes are completed within 50ms in all scenarios. The CDF plot also indicates that the latencies decrease when number of MNs in the mesh topology is increased.

Comparing the simulation results and the experimental results in Figures 5.2 and 5.6 respectively shows that the latencies obtained through simulation are significantly smaller (up to 10 times smaller) than those obtained through experiments. This is because NS2 only

provides for a single channel wireless environment simulation which means that the active scanning process only scans one channel compared to the experimental case where 11 channels are scanned.

5.2.3 Handoff Latency by Using MeshScan

Figure 5.5 (a) and Figure 5.5 (b) show the simulation results for handoff latency under MeshScan during the discovery phase. A dashed reference line shows the fast handoff target of 50 ms.



(a): PDF of the MeshScan

(b): CDF of the MeshScan

Figure 5.7: Simulation MeshScan Handoff Latency under Different Number of MNs

Figure 5.7 (a) shows that the all handoff latencies resulting from the use of MeshScan range from 14ms to 35ms which are less than the target of 50ms. All scenarios exhibit similar delay characteristics apart from the three MNs scenario (i.e. contains a single peak centred around 17ms.). Figure 5.6 (b) shows that approximately 100% of the handoff processes are completed within 50ms for all scenarios. The CDF plot also indicates that the

latencies suffer less delay when number of MNs in the mesh topology is increased. When the number of MNs is greater than four, approximately 95% of the handoff processes are completed within 30ms.

By comparing the MeshScan simulation results with the passive results shown in Figure 5.5 and the active results shown in Figure 5.6, it is clear that MeshScan produces the best handoff performance of the three scan techniques considered. By using the MeshScan scan technique, 100% of the latencies are below the fast handoff target of 50ms for all scenarios considered in this analysis. This compares with 20% and 90% of the latencies for passive and active scanning respectively. Furthermore, the probability of handoff latencies below 30ms is over 80% when using MeshScan compared with 17% and 18% for passive and active scanning respectively.

5.2.4 Simulation Summary

The objective for the simulation work was to verify the feasibility of MeshScan and to compare its latency with that of the standard IEEE 802.11 handoff latency. From the results presented above, The conclusion can been made that MeshScan shows the best performance in finding the next MN for STA to associate with when handoff is required, compared to other scan techniques, namely passive scanning and active scanning. The probability of the latencies below the target 50ms is 100% by using MeshScan, moreover the probability of the latencies below 30ms is over 80% by using MeshScan.

Table 5.2: Compassion of Average Simulation Handoff Latency

Scan Techniques Used\No of MNs	3 MNs	4 MNs	5 MNs	6 MNs
Passive scanning	161.9ms	126.6ms	128.1ms	126.3ms
Active scanning	39.8ms	37.9ms	37.2ms	36.8ms
MeshScan	21.6ms	20.1ms	18.9ms	18.9ms

Table 5.2 summarises the results presented above and shows the average handoff latencies for the three scan techniques when different number of MNs are used in the WMN. It is clear that the latency for handoff process decreases when the number of MNs increases. The latency resulting from the use of MeshScan is approximately 20ms under all the scenarios considered in this analysis which shows MeshScan's potential to become a solution to provide fast handoff in WMNs. Due to the limitation of NS2 as mentioned in chapter 4.2, an experimental analysis was also required to verify the MeshScan performance on a physical WMN testbed.

5.3 MeshScan Prototype Experiments

This section describes experiments that were performed to determine the performance of MeshScan under real operating conditions. Firstly, the comparison of the handoff latency between MeshScan and original Madwifi will be presented to show the performance improvement resulting from the use of MeshScan. Secondly, as discussed in section 3.2.2, the performance test was used to verify the limitation of MeshScan. During the performance experiments, different numbers of MNs were used (between 3 and 6 MNs) and

different background loads of 10 Mbps 15 Mbps and 20 Mbps were introduced. The SmartList was preloaded to client manually before the simulation started in all cases.

5.3.1 Handoff Latency Comparison between MeshScan and Original Madwifi Driver

Figure.5.8 shows handoff latency for the original Madwifi (using active scanning), MeshScan Passive Handoff and MeshScan Active Handoff. The x-axis shows the time in milliseconds and the y-axis shows the normalized frequency of handoff latency. A dashed reference line shows the fast handoff target of 50 ms.

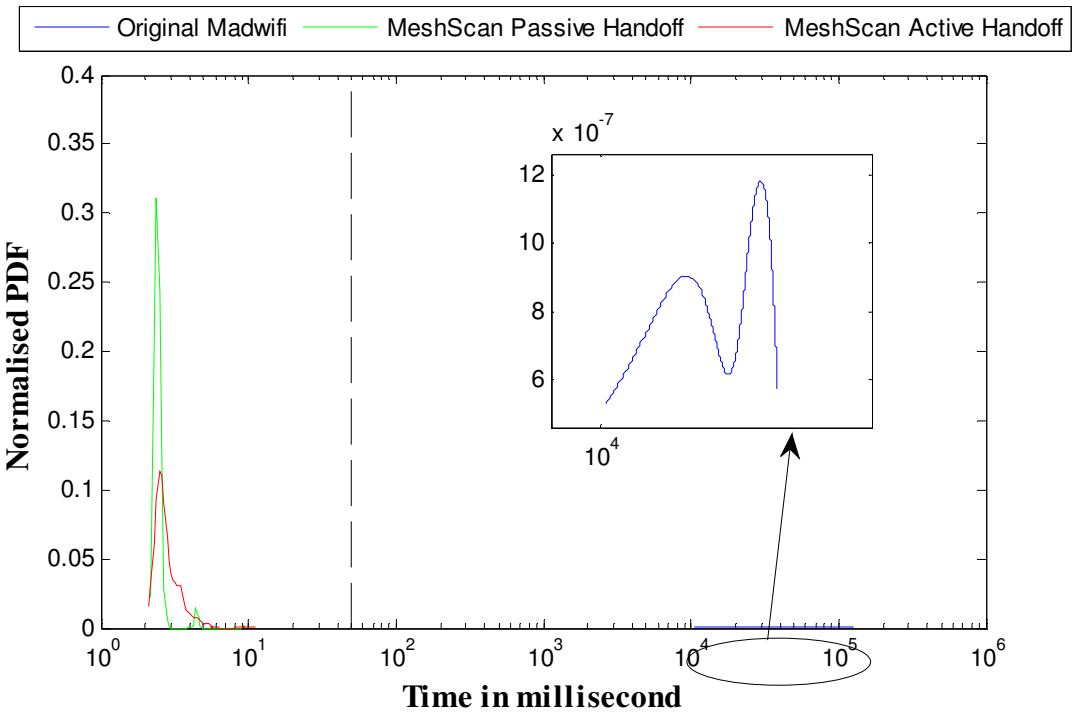


Figure 5.8: PDF of the Handoff Latency

Comparison between MeshScan and Original Madwifi Drive

From Figure 5.8, it can be seen that MeshScan performs well under real network conditions. It can be seen that the handoff latency in the original Madwifi driver appears widely distributed from 10s to 100s of seconds and cannot provide fast handoff. It can also be seen that the handoff latency associated with our MeshScan technique decreases

dramatically under both passive and active handoff where the lowest handoff was just 1.8ms for both passive and active handoff. In the majority of the cases, the handoff latencies were between 1.8 ms to 3 ms when using MeshScan scan technique.

5.3.2 MeshScan Performance Test

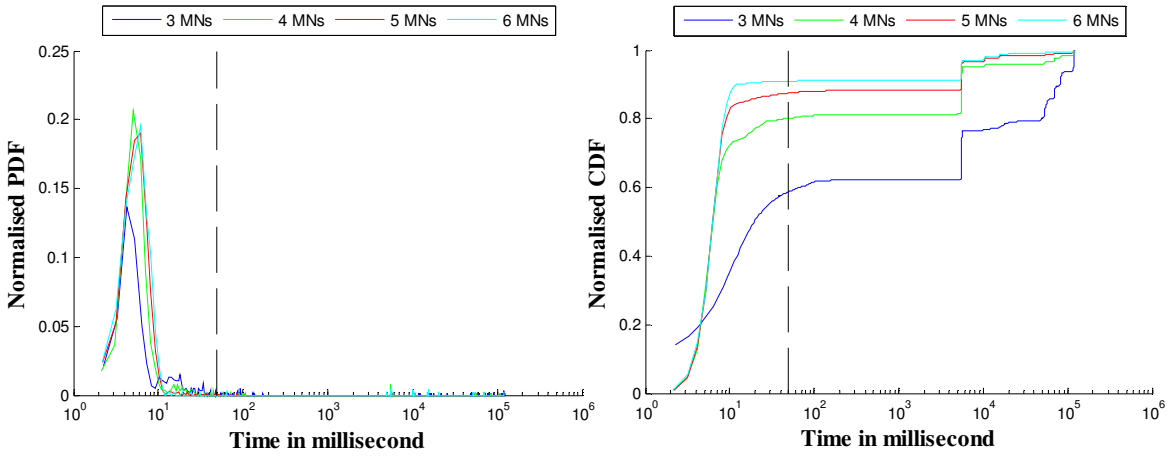
In the previous section, it was shown that MeshScan improves the handoff latency dramatically under no background load conditions. In this section, the further results of the performance tests for the MeshScan mechanism are presented as the network conditions vary. These performance test experiments are divided into two groups which correspond to passive and active handoff. Selected results are presented to show the major findings from experiments, other results can be found in Appendix I.

5.3.2.1 Passive Handoff

Figure 5.9 shows the latency performance of MeshScan (for passive handoff) under a 20 Mbps background load when the number of MNs is increased from three to six. A dashed reference line shows the fast handoff target of 50 ms. Figure 5.9 (a) clearly shows that MeshScan can still operate under a 20 Mbps background load for up to six MNs. The lowest latency is approximately 2.2 ms and most of the handoffs are completed within 10 ms.

From Figure 5.9 (b), it is clear that MeshScan is more efficient when there are more MNs available to the STA. Approximately 60% of the handoffs are completed within 50 ms in the three MNs scenario, while approximately 90% are completed in the six MNs scenario. This is because the MeshScan does not wait for the authentication response frame after transmitting the authentication request frame to a MN, but keeps transmitting authentication request frame to the next MN in the *SmartList*. The MeshScan will stop transmitting

authentication request frame to MNs after either successfully receiving an authentication response frame from a MN or reaching the end of the *SmartList*. The CDF curves for each of the background loads exhibit the same general shape. In particular, the jump in the CDF curve around 5500 ms is due to the completion of the Mesh scan and the reversion to the default active scanning in the Madwifi driver.



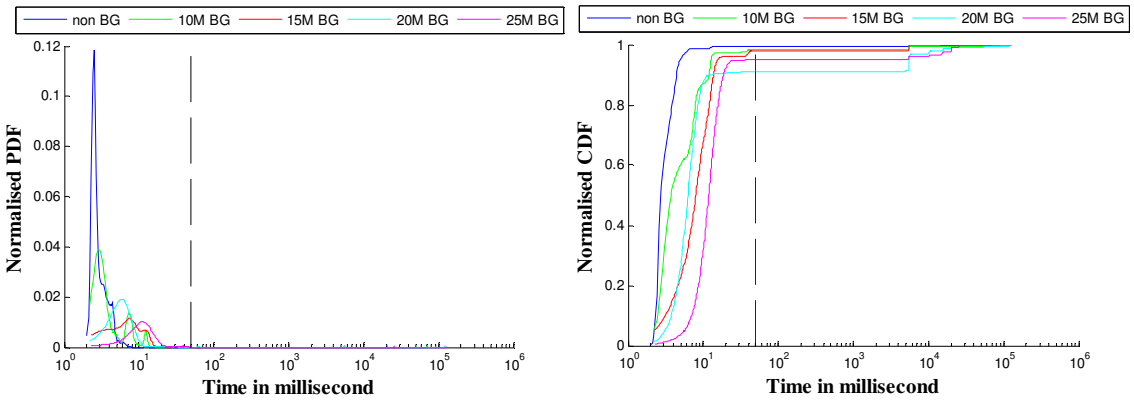
(a) PDF of the Handoff Latency

(b) CDP of the Handoff Latency

Figure 5.9: Experiment MeshScan Passive Handoff Latency under Different Number of MNs

Figure 5.10 shows the performance of MeshScan (passive handoff) under different background loads of 10 Mbps, 15 Mbps, 20 Mbps, and 25 Mbps in a six MNs mesh topology. A dashed reference line shows the fast handoff target of 50 ms. Figure 5.10 (a) shows that that the handoff latency increases when background load increases as expected. The increase in handoff latency when background traffic load was introduced is because the RTT was increased due to network interference and traffic load. From Figure 5.10 (b) it can be seen that when the background traffic load at 20 Mbps, over 90% of the handoffs were

completed within 20 ms and that 90% of the handoffs were completed in 14ms when the background traffic load was 25 Mbps.



(a): PDF of Handoff Latency (b): CDF of Handoff Latency

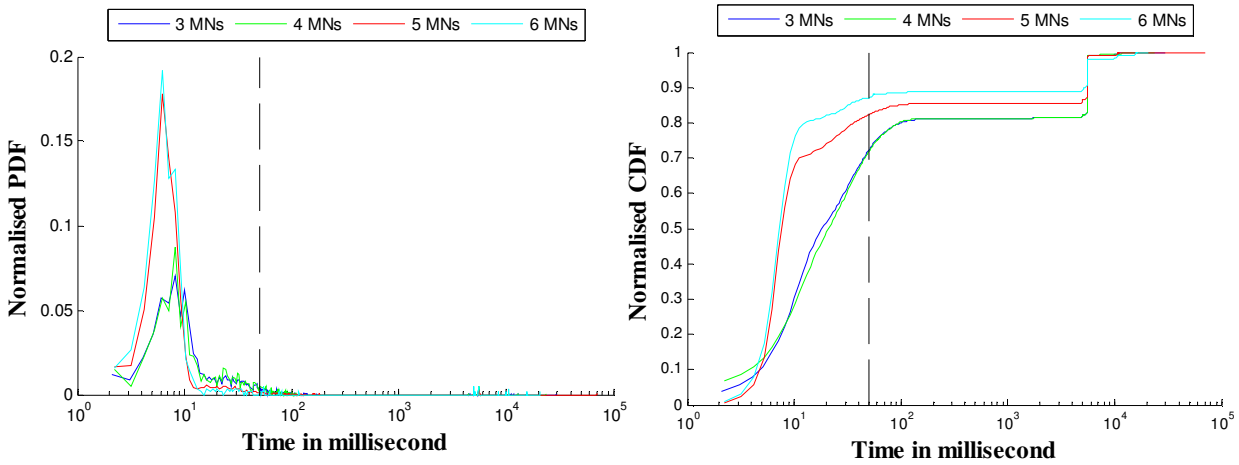
Figure 5.10: Experiment MeshScan Passive Handoff Latency under Different Background Load

5.3.2.2 Active Handoff

Figure 5.11 shows the performance of MeshScan (active handoff) under a 20 Mbps background load when the number of MNs was increased from three to six. A dashed reference line shows the fast handoff target of 50 ms. Figure 5.11 (a) clearly shows that MeshScan still works under a 20 Mbps background load for up to six MNs, where the lowest latency is approximately 2.2 ms and most of handoffs are completed within approximately 14 ms.

From Figure 5.11 (b), it is clear that MeshScan is more efficient when there are more MNs available to the STA. Approximately 70% of the handoffs are completed within 50ms in the three MNs scenario, while approximately 88% are completed in the six MNs scenario.

This is a similar result to the passive handoff case. The CDF curves for each of the background loads exhibit the same general shape. In particular, the jump in the CDF curve around 5500 ms is due to the completion of the Mesh scan and the reversion to the default active scanning in the Madwifi driver.



(a): PDF of Handoff Latency (b): CDF of Handoff Latency

Figure 5.11: Experiment MeshScan Active Handoff Latency under Different Number of MNs

Figure 5.12 shows the performance of MeshScan (active handoff) under different background loads of 10 Mbps, 15 Mbps, 20 Mbps, and 25 Mbps in six MNs mesh topology. A dashed reference line shows the fast handoff target of 50 ms. Figure 5.12 (a) shows that the handoff latency increases when background load increases as expected. The increase in handoff latency when the background traffic load was increased is because the RTT was increased due to network interference and traffic load.

From Figure 5.12 (b) it can be seen that when the background traffic load is 20 Mbps, over 88% of the handoffs were completed within 50 ms and that 76% of the handoffs were completed in 50 ms when the background traffic load was 25 Mbps. It is also clear that the

handoff latency was increased significantly when the background load was 20 Mbps and 25 Mbps compared to when background load was 10 Mbps and 15 Mbps.

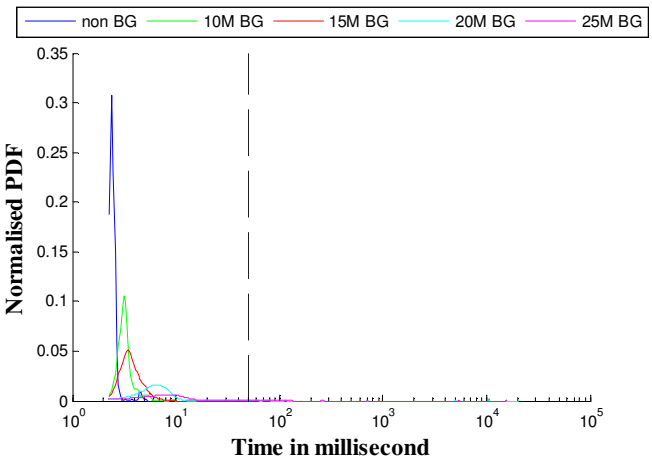


Figure 5.12 (a): PDF of Handoff Latency

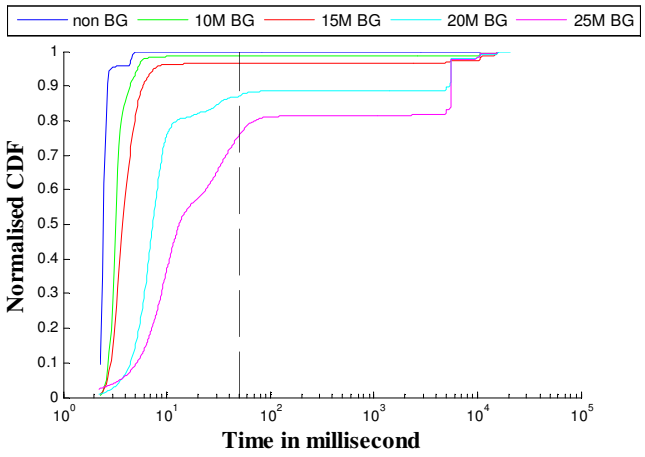


Figure 5.12 (b): CDF of Handoff Latency

Figure 5.12: Experiment MeshScan Active Handoff Latency under Different Background Load

If one compares Figure 5.10 and Figure 5.12, it can be seen that the average handoff latency in passive handoff was less than that in active handoff. Furthermore, the impact on latency in passive handoff is less than in active handoff when background load was introduced. This is due to handoff triggering time (computing time) being measured in active handoff at the client side (an extra $\frac{1}{2}RTT$ time is used for active handoff compared to passive handoff).

5.3.3 Experiment Summary

In this section, the experimental work has demonstrated that MeshScan can operate successfully under real network conditions and it has been shown that the latency associated with handoff can be reduced from seconds to a few milliseconds when no background load is present.

The performance tests show that the MeshScan technique works under heavy background loads up to 25 Mbps when the number of MNs varies between three to six nodes. In passive handoff, over 95% of handoffs were completed within 50 ms under a background load of 25 Mbps and when there were six MNs available. In active handoff, approximately 76% of handoff finished within 50 ms under a background load of 25 Mbps when there were six MNs available. Furthermore, MeshScan is more effective when there are more MNs are available to the STA.

5.4 Chapter Summary

This chapter outlines the results of the three phases of this study: Firstly, the IEEE 802.11 handoff analysis which investigated the delay in each step of the handoff process and determined how much delay each of the phases introduce. This analysis shows that the discovery phase accounts for more than 99% of total handoff latency which can range from a few hundred milliseconds to several seconds. Secondly, the simulation of the MeshScan scheme verified its feasibility. The simulation analysis compared the MeshScan with the IEEE 802.11 standard scanning techniques (passive scanning and active scanning) and the results show that MeshScan yields the best performance of the three scanning techniques. Finally, the MeshScan performance test which further examined the MeshScan performance through experiments under different network conditions. The experiments show that a significant reduction in handoff latency from several seconds to a few milliseconds can be achieved by using MeshScan. Under heavy network load conditions MeshScan still works effectively. For example, under a background load of 25 Mbps, 76% of handoff processes were completed within 50 ms.

These results demonstrate that the MeshScan scheme can provide fast handoff to WMNs in an effective way. The next chapter will outline the conclusion of this study and outline future possible work in this area.

6. Summary and Future Work

6.1 Findings of this Work

Handoff which is a process of managing the connection transition from one MN to another MN in order to maintain network connectivity is becoming a major problem in WMNs. Ideally handoff should be completely transparent to a mobile client when supporting real-time traffic applications such as interactive Voice over IP (VoIP) or video conferencing. The handoff procedure aims to reduce this time as much as possible so that the upper layers (and ultimately the end users) do not notice the connectivity interruption.

However, under the IEEE 802.11 WLAN standard, there are three steps involved in the handoff process in the MAC layer: Discovery, Authentication and Re-association. Previous work [17] has reported that the standard handoff incurs a latency of the order of hundreds of milliseconds to several seconds. Moreover, the discovery step accounts for more than 99% of this latency. Other important issues in handoff are when handoff should be performed and which MN should the client associate with? If the client waits too long to look for new MN then the client may incur a connectivity interruption. If the client is too eager then it may flip back and forth between MNs needlessly (known as ping-ponging) causing network overload.

Therefore, an effective handoff management scheme should be developed to reduce the handoff latency to less than 50 ms, in order to accommodate time critical real-time applications such as VoIP on WMNs.

In this thesis, a practical fast handoff management scheme have been developed called MeshScan, to manage when handoff should be performed and which MN the client should

associate with. Theoretically, MeshScan can reduce the latency associated with handoff by using open system authentication where no key exchange is involved.

MeshScan provides a novel usage of the open system authentication phase to reduce channel scanning latency in both passive handoff and active handoff. MeshScan maintains a list of MNs in a *SmartList* and performs unicast scanning by transmitting authentication request frames to discover available MNs. It then performs handoff instead of broadcasting probe request frames. Consequently MeshScan is fully compatible with all the IEEE 802.11 standards, in particular it is compatible with the recent IEEE 802.11r standard developed for supporting fast handoff. MeshScan addresses when and where a STA will handoff to under the discovery phase. IEEE 802.11r provides the Fast BSS Transition mechanism to establish a connection with a MN under the authentication and association phases.

A set of computer and experimental studies were conducted in order to investigate the performance of the MeshScan fast handoff scheme in an IEEE 802.11 WMN when the number of MNs is increased and when background traffic is introduced. The studies can be divided up into two main groups: computer simulations and experiments

In the computer simulations, NS2 was used to implement the theoretical procedures of the MeshScan and to simulate MeshScan under different network scenarios in order to verify the feasibility of MeshScan.

In the experiments, the Madwifi driver was used to develop a prototype of the MeshScan which was able to be run on a Linux platform. A set of experiments were conducted to analyze the performance MeshScan under different network conditions in the CNRI's mesh network. The experimental testing scenarios were divided into two main categories comprising passive handoff and active handoff.

Over the course of experimentation, the effectiveness of our scheme was demonstrated by comparing it to the IEEE 802.11 standard handoff latency and other fast handoff schemes. The following main observations were made.

- Both passive scanning and active scanning are not suitable for implementing the fast handoff scheme in WMNs
- MeshScan scheme addresses the core problem of when handoff should occur and which MN to handoff to?
- MeshScan can reduce the handoff delay significantly from several seconds to a minimum of 2 milliseconds which represents a reduction of over 99%.
- MeshScan will continue to operate under heavy background loads on the network.
- The more MNs that are available to the client STA, the more efficiently that MeshScan operates

The MeshScan fast handoff scheme has been shown to produce a significant reduction in the handoff latency from several seconds to minimum of 2 milliseconds in the absence of any background traffic. Under heavy background load conditions (i.e. for a 25 Mbps background traffic load) it was shown that 75% of the handoff processes were completed within 50 milliseconds which is the upper limit permitted for seamless handoff for VoIP applications. Compared to the standard handoff scheme, this represents an improvement of approximately 99%. Also MeshScan has been shown to be compatible with the recent IEEE 802.11r standard which has been developed to further improve the handoff latency.

6.2 Future Work

In this work a client side fast handoff scheme for WMNs called MeshScan has been developed and analyzed. Although this scheme has been shown to dramatically improve handoff latency in WMN, further analysis of the scheme under different network conditions could be performed.

There are some limitations that should be pointed out concerning the experimental setup. Due to the facility environment, all the MNs and client STA were operating in channel 60, under the 802.11a mode in order to realise a clean wireless medium for our experiments. Consequently, no channel switching was required during the handoff process. Further research may examine MeshScan in a multi-channel (non-overlapped and overlapped) mesh testbed. In addition, the client STA had a fixed location in each experiment. Therefore, it would be useful to examine the performance of MeshScan when the client STA moves at different speeds and in different environment scenarios. (i.e. open space, office and multiple MeshScan users etc.). Further research may also include determining the overall performance improvement when MeshScan is combined with the recent IEEE 802.11r standard.

From the technical point of view, the MeshScan does not concern itself with QoS in the handoff process which means that although MeshScan allows a STA to quickly handoff from one MN to another, it does not guarantee the link quality. (i.e. throughput, link rate and available bandwidth etc.). Another important consideration for MeshScan is that MeshScan relies on a list of MNs which is given or cached on the STA. This means the STA needs to learn or be given the list in order that MeshScan can function immediately when the STA joins new WMNs. Therefore, further research can be carried out in this area

in order to develop a new distributed network assisted fast handoff protocol. The protocol should enable a mesh node to dynamically generate a list of active mesh nodes. Mesh nodes can then actively deliver the list to assist a client's handoff process and thereby eliminate the continuity problem [2-3] for VoWi-Fi users

In conclusion, an efficient and powerful client-side technique have been developed called MeshScan. This technique addresses the core problem of when handoff should occur and which MN to handoff to in MAC layer. The feasibility of MeshScan to significantly support fast handoff in WMNs has been demonstrated through extensive computer simulations and experiments. The results show that MeshScan has ability to dramatically reduce the standard latency from seconds to milliseconds and can operate under heavy background load conditions (e.g. 76% of handoffs were completed within 50 ms under a 25 Mbps background load). Also MeshScan is fully compatible with new IEEE 802.11r which addresses fast handoff from the perspective of QoS and security, which balances the impact of authentication based on 802.11i.

Bibliography

1. In-Stat. *VoIP Penetration Forecast to Reach 79% of US Businesses by 2013*. 2010 Available from: http://www.in-stat.com/press.asp?ID=2720&sku=IN1004350CT. (Last checked on 28th April 2010)

2. Kelly, E.B. *Quality of Service In Internet Protocol (IP) Networks*. 2002; Available from: http://www.wainhouse.com/files/papers/wr-qos-in-ip-networks.pdf. (Last checked on 28th April 2010)

3. Networks, D. *Voice Quality beyond IP QoS*. 2007; Available from: http://www.ditechnetworks.com/learningcenter/whitepapers/WP_VoIP_Voice_Quality.pdf. (Last checked on 28th April 2010)

4. *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Fast Basic Service Set (BSS).* IEEE Std 802.11r-2008 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008), 2008: p. c1-108.

5. Ye, Y., et al. *A Dual Re-Authentication Scheme for Fast Handoff in IEEE 802.11 Wireless Mesh Networks*. in *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*. 2009. Budapest, Hungary.

6. Mustafa, N., et al. *Pre-scanning and dynamic caching for fast handoff at MAC layer in IEEE 802.11 wireless LANs*. in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*. 2005. Washington DC, USA.

7. Ramani, I. and S. Savage. *SyncScan: practical fast handoff for 802.11 infrastructure networks*. in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. 2005. Miami, FL, USA.

8. I. F. Akyildiz, X.W., and W. Wang, *Wireless mesh networks: a survey.* Computer Networks, 2005. **47**(4): p. 445-487.

9. Akyildiz, I.F., X. Jiang, and S. Mohanty, *A survey of mobility management in next-generation all-IP-based wireless systems.* IEEE Wireless Communications, 2004. **11**(4): p. 16-28.

10. Wei, L., Z. Qing-An, and D.P. Agrawal. *A reliable active scanning scheme for the IEEE 802.11 MAC layer handoff*. in *Radio and Wireless Conference, 2003. RAWCON '03. Proceedings*. 2003.

11. Yuh-Shyan, C., C. Chung-Kai, and C. Ming-Chin. *DeuceScan: Deuce-Based Fast Handoff Scheme in IEEE 802.11 Wireless Networks*. in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*. 2006. Montréal, Québec, Canada.

12. Rivera, N., *Seamless connectivity and mobility in wireless mesh networks*. 2009, Johns Hopkins University. p. 90.

13. Zhang, Z., R.W. Pazzi, and A. Boukerche, *Mobility management protocols for wireless mesh networks*, in *Proceedings of the 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*. 2009, ACM: Tenerife, Canary Islands, Spain. p. 107-110.

14. Yan, Y., et al., *A dual re-authentication scheme for fast handoff in IEEE 802.11 wireless mesh networks*, in *Proceedings of the 2009 IEEE conference on Wireless Communications \& Networking Conference*. 2009, IEEE Press: Budapest, Hungary. p. 2186-2190.

15. Chen, Y., K. Kowalik, and M. Davis, *MeshScan: fast and efficient handoff in IEEE802.11 mesh networks*, in *Proceedings of the 7th ACM international symposium on Mobility management and wireless access*. 2009, ACM: Tenerife, Canary Islands, Spain. p. 105-108.

16. Chen, Y., K. Kowalik, and M. Davis. *MeshScan: Performance of Passive Handoff and Active Handoff*. in *Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on*. 2009. Nanjing , China.

17. Jain, A. *Handoff Delay for 802.11b Wireless LANs*. 2003; Available from: http://www.docstoc.com/docs/22718324/Handoff-Delay-for-80211b-Wireless-LANs/.(Last checked on 28th April 2010)

18. Ye, Y., C. Hua, and S. Seung-Woo. *Performance Analysis of IEEE802.11 Wireless Mesh Networks*. in *Communications, 2008. ICC '08. IEEE International Conference on*. 2008. Beijing, China.

19. *IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE Std 802.11-1997, 1997: p. i-445.

20. *Wi-Fi Alliance*. Available from: http://www.wi-fi.org/.

21. *Supplement to IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band.* IEEE Std 802.11b-1999, 2000: p. i-90.

22. *Draft Supplement to Standard [for] Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz band (Amendment to IEEE Std 802.11, 1999 Edition).* IEEE Std P802.11g/D8.2, 2003.

23. *Supplement to IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band.* IEEE Std 802.11a-1999, 1999: p. i.

24. *IEEE Draft Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific equirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment : Enhancements for Higher Throughput.* IEEE Unapproved Draft Std P802.11n/D8.0, Feb 2009, 2009.

25. *Approved Draft IEEE Standards for Local and Metropolitan Area Networks: Port-Based Network Access Control Replaced by 802.1X-2004.* IEEE Std P802.1X-REV/D11, 2004.

26. Pathak, A., A. Mangalam Srivatsa, and X. Jiang. *An Analytical Model for Handoff Overhead Analysis in Internet-Based Infrastructure Wireless Mesh Networks*. in *Communications, 2008. ICC '08. IEEE International Conference on*. 2008. Beijing, China.

27. *IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* ANSI/IEEE Std 802.11, 1999 Edition (R2003), 2003: p. i-513.

28. Velayos, H. and G. Karlsson. *Techniques to reduce the IEEE 802.11b handoff time*. in *Communications, 2004 IEEE International Conference on*. 2004. Paris, France.

29. Mishra, A., M. Shin, and W. Arbaugh, *An empirical analysis of the IEEE 802.11 MAC layer handoff process.* SIGCOMM Comput. Commun. Rev., 2003. **33**(2): p. 93-102.

30. Xinrui. Wang, T.-F.L., Lei. Chen, *Synchronization and Time Resolution Improvement for 802.11 WLAN OWPT Measurement*, in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009*. 2009: Hong Kong.

31. Gurpal Singh, A.P.S.A.a.B.S.S., *Mobility Management Technique for Real Time Traffic in 802.11 Networks.* Journal of Computer Science 2007. **3**(6): p. 390-398.

32. Shin, S., et al., *Reducing MAC layer handoff latency in IEEE 802.11 wireless LANs*, in *Proceedings of the second international workshop on Mobility management \& wireless access protocols*. 2004, ACM: Philadelphia, PA, USA. p. 19-26.

33. Tang, D. and M. Baker, *Analysis of a metropolitan-area wireless network.* Wireless Networks, 2002. **8**(2/3): p. 107-120.

34. Chambers, B. *The grid roofnet: a rooftop ad hocwireless network.* 2002; Available from: http://pdos.csail.mit.edu/papers/grid:bac-meng.pdf. (Last checked on 28th April 2010)

35. Bicket, J., et al., *Architecture and evaluation of an unplanned 802.11b mesh network*, in *Proceedings of the 11th annual international conference on Mobile computing and networking*. 2005, ACM: Cologne, Germany. p. 31-42.

36. Camp, J.D., E.W. Knightly, and W.S. Reed, *Developing and deploying multihop wireless networks for low-income communities.* Journal of Urban Technology, 2006. **13**(3): p. 129-137.

37. *The Champaign-Urbana community wireless network.* Available from: http://www.cuwin.net. (Last checked on 28th April 2010)

38. *Microsoft Research Networking Research Group* Available from: http://research.microsoft.com/mesh. (Last checked on 28th April 2010)

39. Adya, A., et al., *A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks*, in *Proceedings of the First International Conference on Broadband Networks*. 2004, IEEE Computer Society. p. 344-354.

40. Draves, R., J. Padhye, and B. Zill, *Routing in multi-radio, multi-hop wireless mesh networks*, in *Proceedings of the 10th annual international conference on Mobile computing and networking*. 2004, ACM: Philadelphia, PA, USA. p. 114-128.

41. Ramachandran, K.N., et al., *On the design and implementation of infrastructure mesh networks*, in *IEEE Workshop on Wireless Mesh Networks*. 2005: Santa Clara, CA, USA.

42. Navda, V., A. Kashyap, and S.R. Das. *Design and evaluation of iMesh: an infrastructure-mode wireless mesh network.* in *World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a*. 2005. Taormina, Giardini Naxos.

43. Ganguly, S., et al., *Performance Optimizations for Deploying VoIP Services in Mesh Networks.* IEEE Selected Areas in Communications, 2006. **24**(11): p. 2147-2158.

44. Amir, Y., et al., *Fast handoff for seamless wireless mesh networks*, in *Proceedings of the 4th international conference on Mobile systems, applications and services*. 2006, ACM: Uppsala, Sweden. p. 83-95.

45. Amir, Y., et al. *An Inter-domain Routing Protocol for Multi-homed Wireless Mesh Networks*. in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*. 2007. Helsinki, Finland.

46. IETF, *IP Mobility Support for IPv4*. 2006.

47. Buddhikot, M., et al., *MobileNAT: a new technique for mobility across heterogeneous address spaces.* Mob. Netw. Appl., 2005. **10**(3): p. 289-302.

48. Sun, Y., E. Belding-Royer, and C. Perkins, *Internet connectivity for ad hoc mobile networks.* International Journal of Wireless Information Networks, 2002.

49. Yuan, S. and E.M. Belding-Royer. *Application-oriented routing in hybrid wireless networks*. in *Communications, 2003. ICC '03. IEEE International Conference on*. 2003. Anchorage Convention Center, Anchorage, AK

50. Matthew, W., J. Miller, and N. Vaidya. *A hybrid network implementation to extend infrastructure reach*. 2003; Available from: http://users.crhc.illinois.edu/nhv/papers/hybrid-tech.pdf. (Last checked on 28th April 2010)

51. Jonsson, U., et al. *MIPMANET-mobile IP for mobile ad hoc networks*. in *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*. 2000. Boston, Massachusetts, USA.

52. Yu-Chee, T., S. Chia-Ching, and C. Wen-Tsuen, *Integrating mobile IP with ad hoc networks.* Computer, 2003. **36**(5): p. 48-55.

53. Ratanchandani, P. and R. Kravets. *A hybrid approach to Internet connectivity for mobile ad hoc networks*. in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*. 2003. New Orleans, LA, USA

54. Andr\, et al., *Cellular IP: a new approach to Internet host mobility.* SIGCOMM Comput. Commun. Rev., 1999. **29**(1): p. 50-65.

55. Ramjee, R., et al. *HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks*. in *Network Protocols, 1999. (ICNP '99) Proceedings. Seventh International Conference on*. 1999. Toronto, Canada

56. Campbell, A.T., et al., *Comparison of IP micromobility protocols.* IEEE Wireless Communications, 2002. **9**(1): p. 72-82.

57. DaSilva, L.A., et al., *The resurgence of push-to-talk technologies.* IEEE Communications Magazine, 2006. **44**(1): p. 48-55.

58. Grilo, A., P. Estrela, and M. Nunes, *Terminal independent mobility for IP (TIMIP).* IEEE Communications Magazine, 2001. **39**(12): p. 34-41.

59. Sharma, S., Z. Ningning, and C. Tzi-cker, *Low-latency mobile IP handoff for infrastructure-mode wireless LANs.* IEEE Selected Areas in Communications, 2004. **22**(4): p. 643-652.

60. Das, S., et al., *IDMP: an intradomain mobility management protocol for next-generation wireless networks.* IEEE Wireless Communications, 2002. **9**(3): p. 38.

61. Hsieh, R., Z.G. Zhou, and A. Seneviratne. *S-MIP: a seamless handoff architecture for mobile IP*. in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. 2003. San Francisco, California, USA.

62. Soliman, K.M.H., C. Castelluccia, and L. Bellier, *Hierarchical mobile ipv6 mobility management (hmipv6)*. 2004.

63. Ram\, et al., *Fast and scalable wireless handoffs in supports of mobile Internet audio.* Mob. Netw. Appl., 1998. **3**(4): p. 351-363.

64. Helmy, A.A.G., M. Jaseemuddin, and G. Bhaskara, *Multicast-based mobility: a novel architecture for efficient micromobility.* IEEE Selected Areas in Communications, 2004. **22**(4): p. 677-690.

65. Forte, A.G. and H. Schulzrinne. *Cooperation Between Stations in Wireless Networks*. in *Network Protocols, 2007. ICNP 2007. IEEE International Conference on*. 2007. Beijing, China.

66. Bejerano, Y., I. Cidon, and J. Naor, *Efficient handoff rerouting algorithms: a competitive on-line algorithmic approach.* IEEE/ACM Networking, 2002. **10**(6): p. 749-760.

67. Chiasserini, C.F. and R. Lo Cigno, *Handovers in wireless ATM networks: in-band signaling protocols and performance analysis.* IEEE Wireless Communications, 2002. **1**(1): p. 87-100.

68. Vatn, J.-O. *An experimental study of IEEE 802.11b handover performance and its effect on voice traffic*. 2003; Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.5771&rep=rep1&type=pdf. (Last checked on 28th April 2010)

69. Brik, V., A. Mishra, and S. Banerjee, *Eliminating handoff latencies in 802.11 WLANs using multiple radios: applications, experience, and evaluation*, in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. 2005, USENIX Association: Berkeley, CA. p. 27-27.

70. Sunggeun, J., C. Munhwan, and C. Sunghyun, *Multiple WNIC-based handoff in IEEE 802.11 WLANs.* IEEE Communications Letters, 2009. **13**(10): p. 752-754.

71. Bangolae, S., C. Bell, and E. Qi, *Performance study of fast BSS transition using IEEE 802.11r*, in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*. 2006, ACM: Vancouver, British Columbia, Canada. p. 737-742.

72. Chung-Sheng, L., T. Yung-Chih, and C. Han-Chieh. *A Neighbor Caching Mechanism for Handoff in IEEE 802.11 Wireless Networks*. in *Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on*. 2007.

73. *CNRI Wireless Mesh Testbed*. Available from: http://www.cnri.dit.ie/research.mesh.testbed.html. (Last checked on 28th April 2010)

74. *Network Simulator v2*. Available from: http://www.isi.edu/nsnam/ns/.(Last checked on 28th April 2010)

75. *MadWifi - a Linux kernel device driver for Wireless LAN chipsets from Atheros*. Available from: http://madwifi-project.org/.(Last checked on 28th April 2010)

76. *Atheros*. Available from: http://www.atheros.com/pt/index.html. (Last checked on 28th April 2010)

77. *Hardware Abstraction Layer*. Available from: http://www.freedesktop.org/wiki/Software/hal. (Last checked on 28th April 2010)

78. *D-ITG Tool* Available from: http://www.grid.unina.it/software/ITG/.(Last checked on 28th April 2010)

79. *Wireshark* Available from: http://www.wireshark.org/.(Last checked on 28th April 2010)

80. *Network Simulator 2 Documentation*. Available from: http://www.isi.edu/nsnam/ns/doc/index.html. (Last checked on 28th April 2010)

81. *Madwifi Driver Summary* Available from: http://mesh.calit2.net/whzhao/madwifi_summary.pdf. (Last checked on 28th April 2010)

82. Ferre, P., et al. *Throughput analysis of IEEE 802.11 and IEEE 802.11e MAC*. in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*. 2004. Atlanta, GA USA.

83. Shao-Cheng, W. and A. Helmy. *Performance Limits and Analysis of Contention-based IEEE 802.11 MAC*. in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. 2006. Tampa, Florida, USA.

Appendices I - (Testbed Setup)

IMPORTANT: Default value applied if any parameter did not mention in the Tables below.

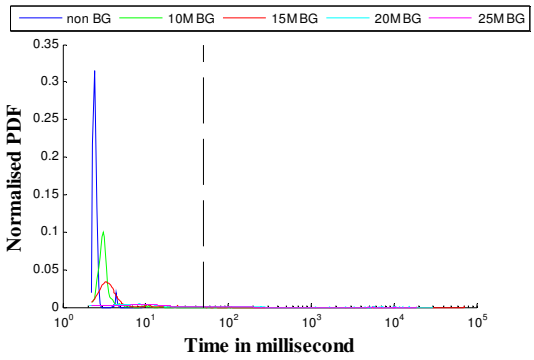
NS2 Simulation Parameters			
Operation Mode	802.11 a	*Authentication*	Open system Key
Operation Channel	Channel 60 (5.32 GHz)	*packet_size*	1Kb
Radio Propagation Type	Propagation/TwoRayGround	*gap_size*	0.001
Network Interface Type	Phy/WirelessPhy	*SIFS*	10µs
Interface Queue Type	Queue/DropTail/PriQueue	*SlotTime*	20µs
Antenna Model	Antenna/OmniAntenna	*CWmin*	31
Max Packet In ifq	50	*CWmax*	1023
Transmission Power	0.025	dataRate	11Mbps
Reception Threshold	5.82916e-09	*basicRate*	1Mbps
Carrier Sensing Threshold	5.24624e-09	*RTSThreshold*	30000
Receive Antenna Gain	1.0	*Link Layer Type*	LL
Transmit Antenna Gain	1.0	*MAC Type*	Mac/802_11
System Loss Factor	1.0	*Routing Protocol*	DSDV

Madwifi Experiment Parameters			
Operation Mode	802.11 a	*Network Type*	Ad-hoc
Operation Channel	Channel 60 (5.32 GHz)	*ATH_TXMAXTRY*	1
IEEE80211_TRANS_WAIT	1ms	essid	*mesh_handoff*
Authentication	Open system Key	*Data Rate*	11Mbps

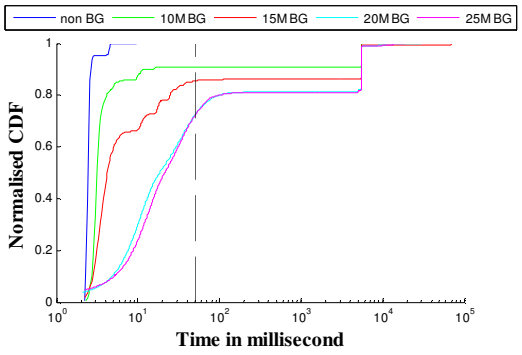
Appendices II - (Full Results)

Active Handoff MeshScan

3 MNs under Different BG load

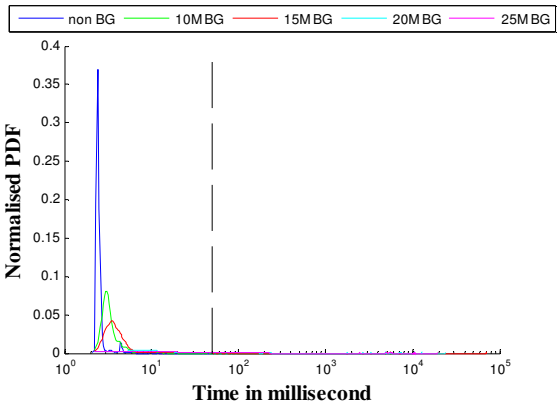


PDF

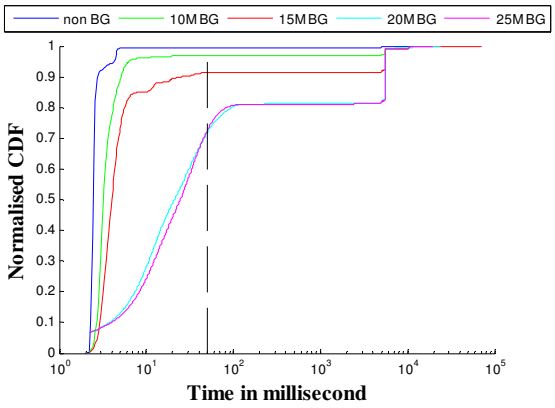


CDF

4 MNs under Different BG load

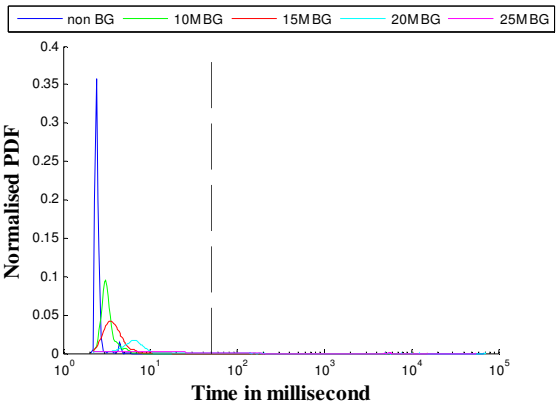


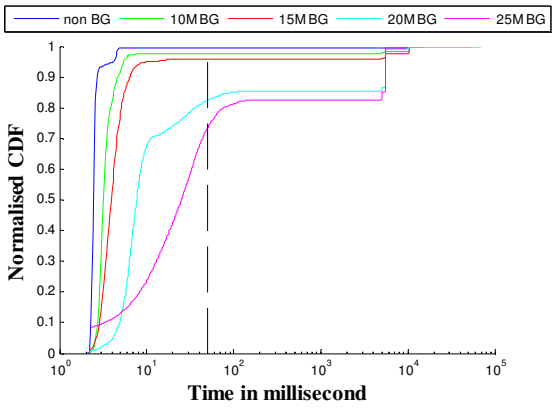
PDF



CDF

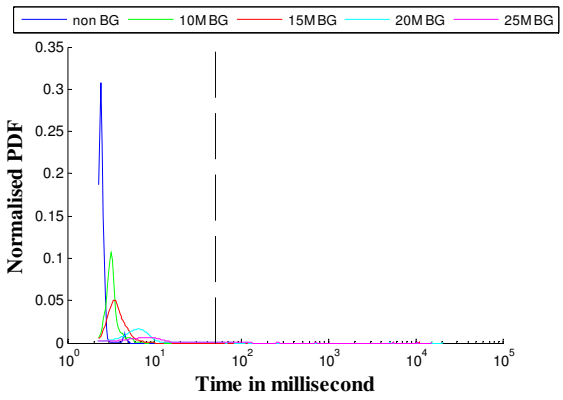
5 MNs under Different BG load

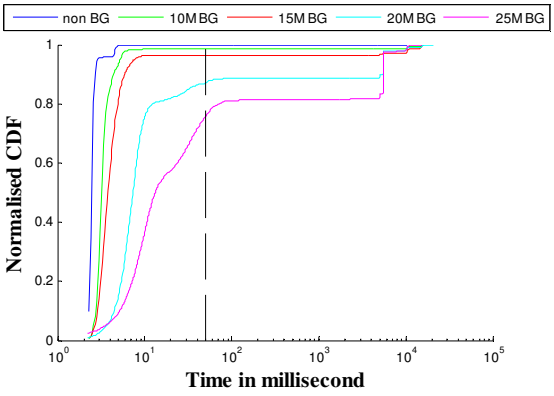




PDF CDF

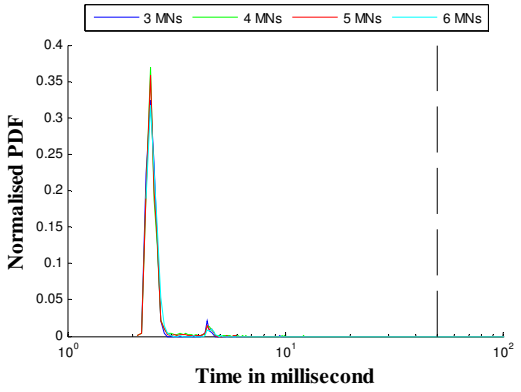
6 MNs under Different BG load

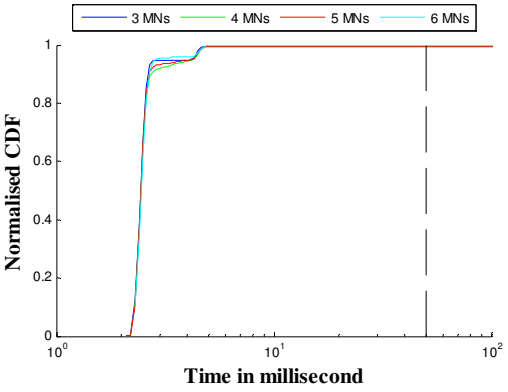




PDF CDF

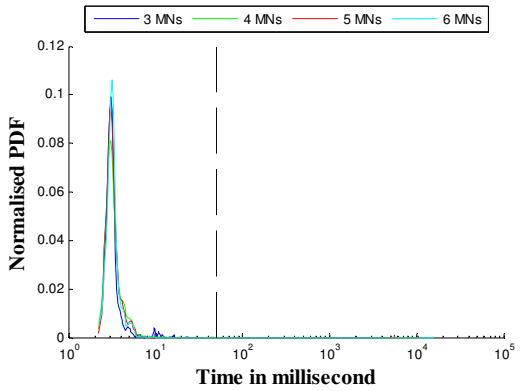
Different number of MNs under no BG Load

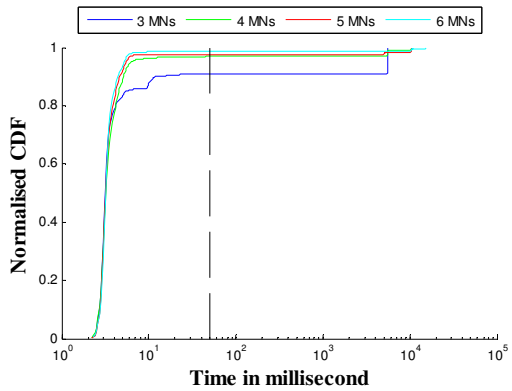




PDF CDF

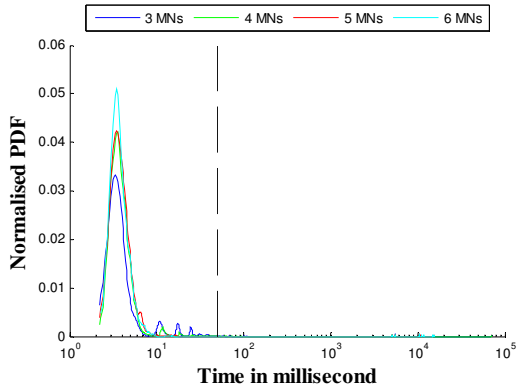
Different number of MNs under 10M BG Load



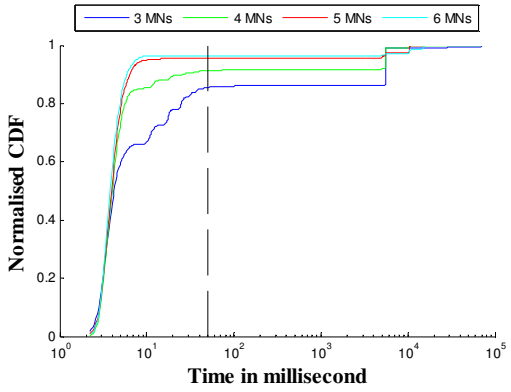


PDF CDF

Different number of MNs under 15M BG Load

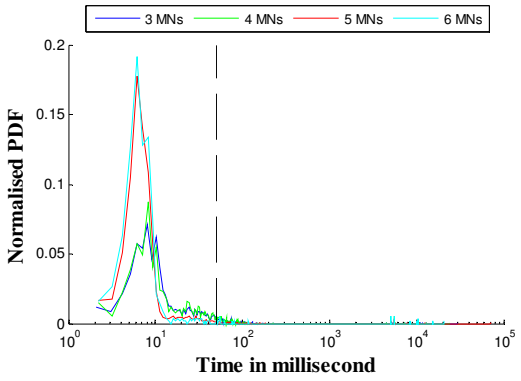


PDF

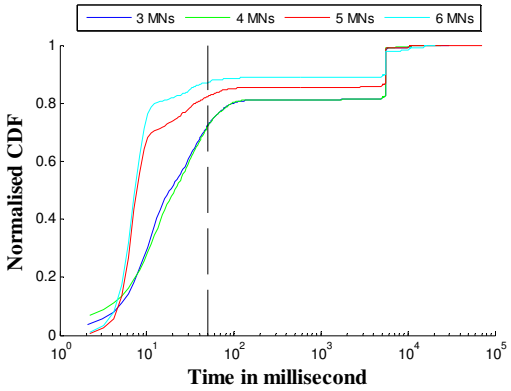


CDF

Different number of MNs under 20M BG Load

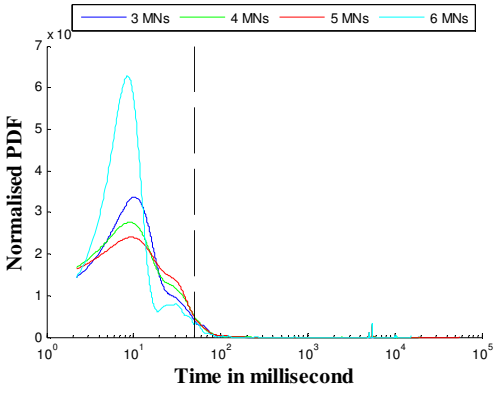


PDF

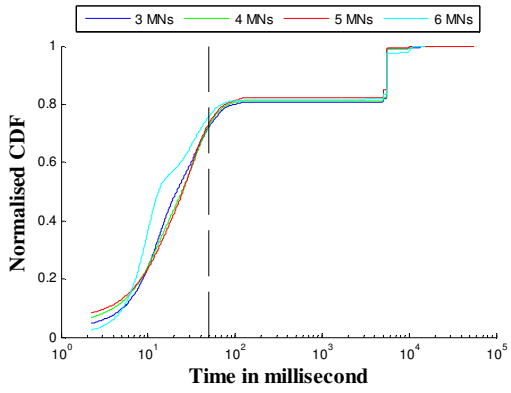


CDF

Different number of MNs under 25M BG Load



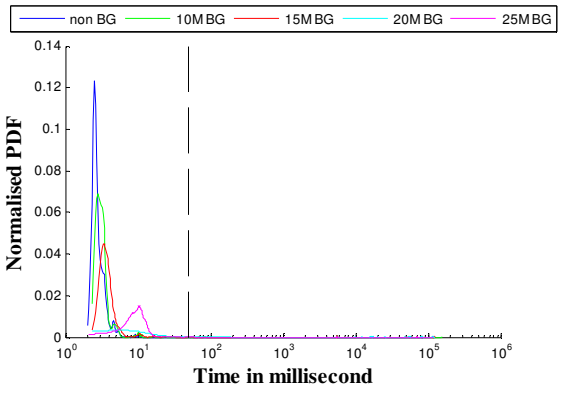
PDF



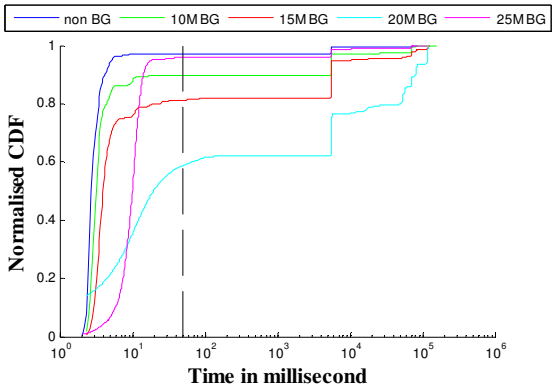
CDF

Passive Handoff MeshScan

3 MNs under Different BG load

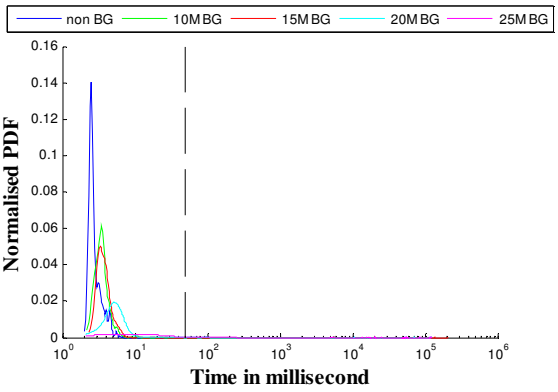


PDF

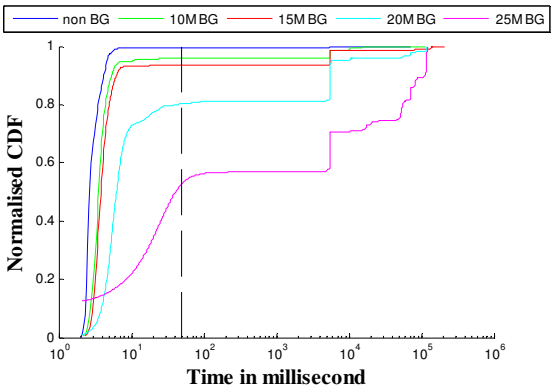


CDF

4 MNs under Different BG load

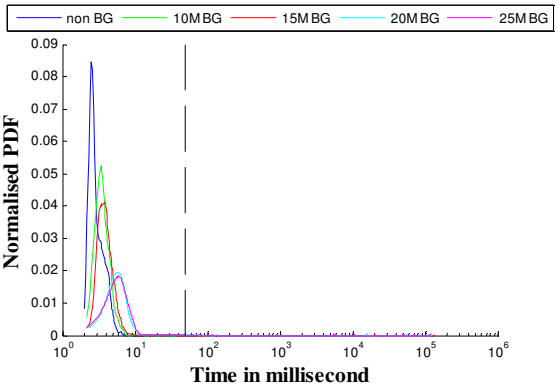


PDF

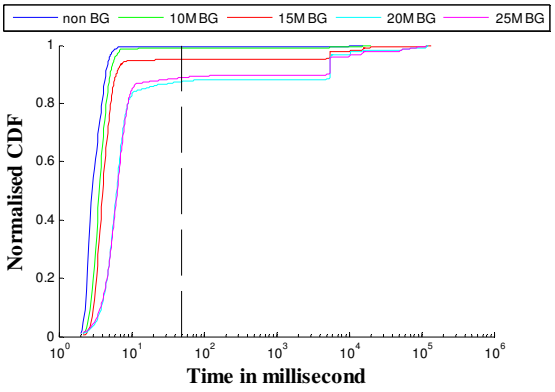


CDF

5 MNs under Different BG load

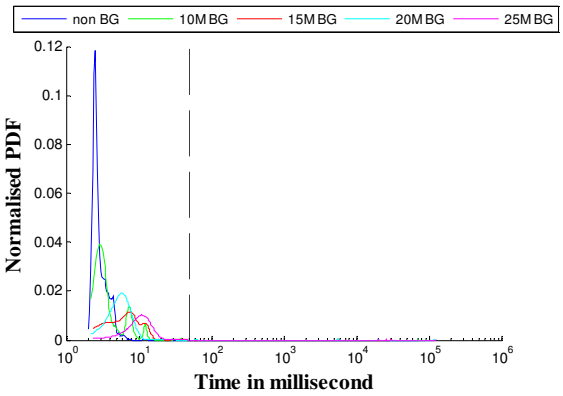


PDF

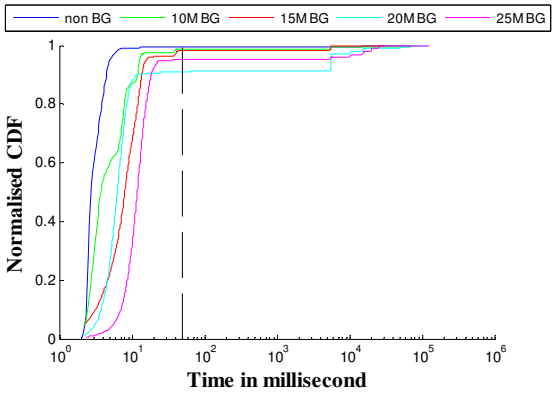


CDF

6 MNs under Different BG load

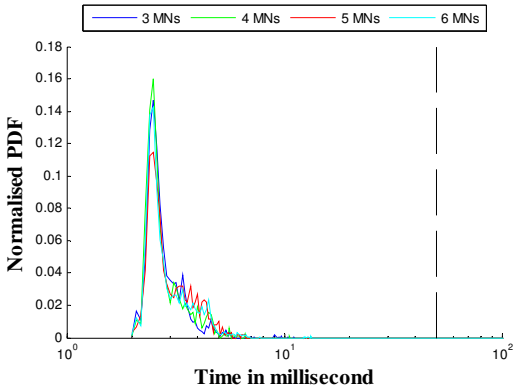


PDF

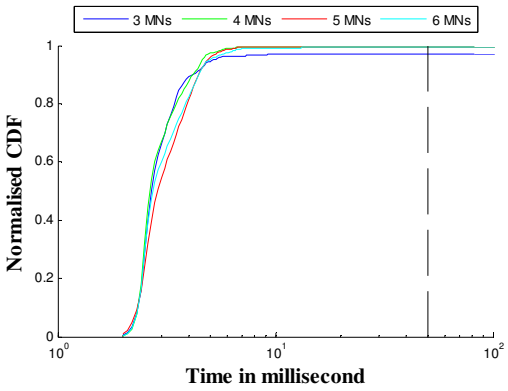


CDF

Different number of MNs under no BG Load

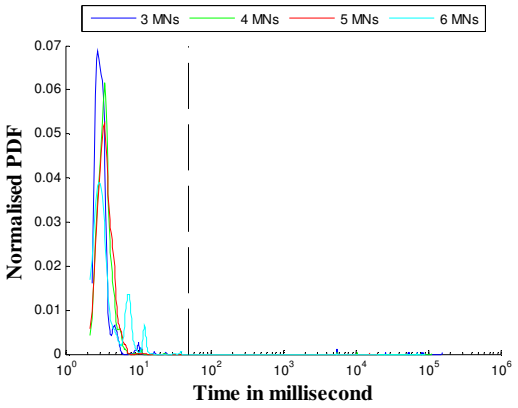


PDF

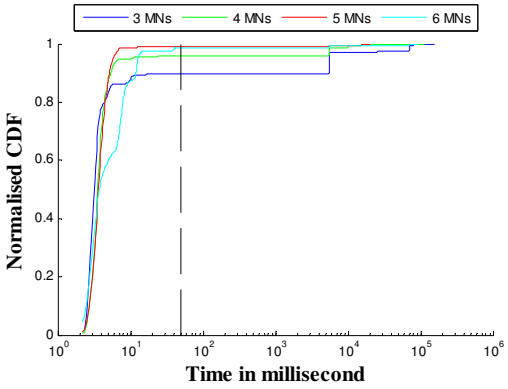


CDF

Different number of MNs under 10M BG Load

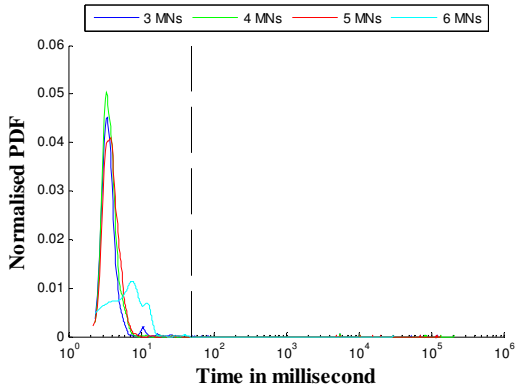


PDF

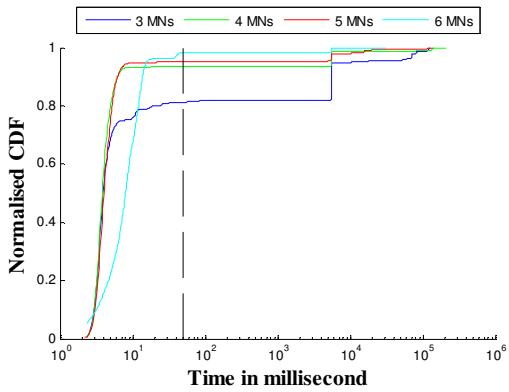


CDF

Different number of MNs under 15M BG Load

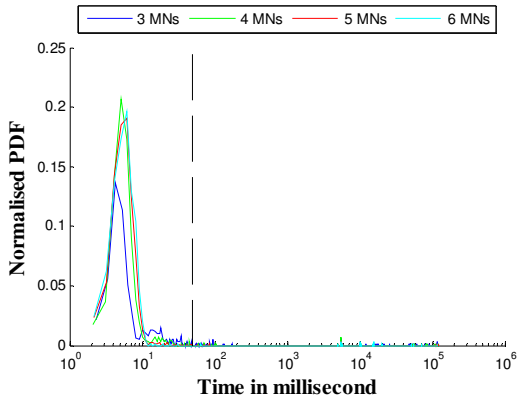


PDF

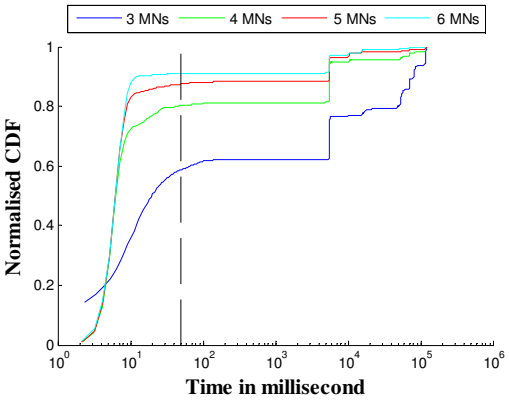


CDF

Different number of MNs under 20M BG Load

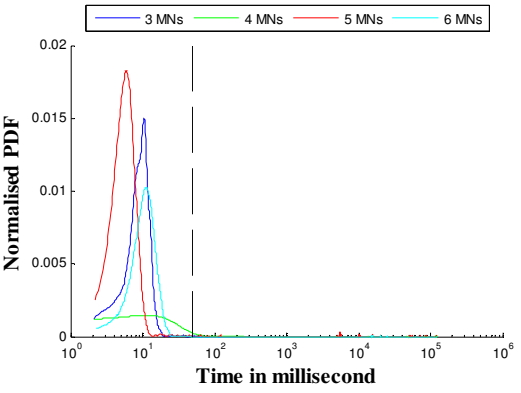


PDF

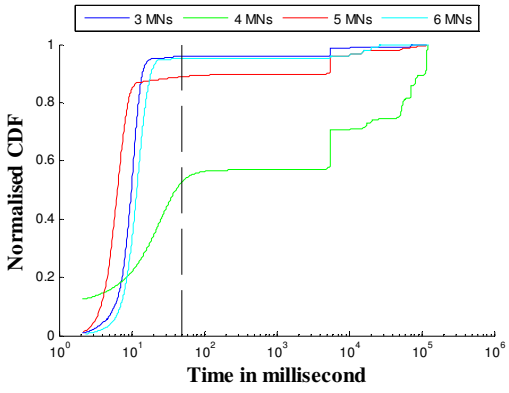


CDF

Different number of MNs under 25M BG Load



PDF



CDF

List of publications

- MeshScan: Fast and Efficient Handoff in IEEE802.11 Mesh Networks

  The 7th ACM International Symposium on Mobility Management and Wireless Access (MobiWAC 2009), Tenerife, Canary Islands, Spain, October 2009.

- MeshScan: Performance of Passive Handoff and Active Handoff

  International Conference on Wireless Communication & Signal Processing (WCSP'09), Nanjing, China, November 2009.