

**SIMULATION OF MEDIA INDEPENDENT
HANDOVER ACROSS HETEROGENEOUS
NETWORKS (802.21) IN NS2**

A PROJECT REPORT

Submitted by

K. K. DHEEPAK	20023224
S. MADHAN	20023245
I. MOHAMMED SHAREEF	20023248

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING



DEPARTMENT OF ELECTRONICS ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY

CHENNAI - 600044

APRIL 2006

ACKNOWLEDGEMENT

We are extremely glad in expressing our sense of gratitude to our Dean Dr.P. Kanagasabapathy for his support.

We are thankful to Dr. J. Shanmugam, Professor & Head, Department of Electronics Engineering, Madras Institute of Technology, for facilitating this project work in the Department.

We are very glad in expressing our sense of gratitude to our guides Ms. G. Sumithra, Lecturer, Department of Electronics Engineering, Madras Institute of Technology and Jackson Juliet Roy, AU-KBC Research centre, Madras Institute of Technology for their valuable suggestions and guidance rendered to us during the course of the work.

We are also very grateful to the Network Project Panel Members, Dr. V. Vaidehi, Mrs. S. Indiragandhi and Mr. A. Velmurugan for their valuable suggestions during the Project Reviews.

We would also like to extend our sincere thanks to all our friends whose constant encouragement and support helped us to complete this phase of our project successfully.

We express our deepest gratitude to our parents for having given this opportunity and having made us what we are.

K. K. Dheepak	20023224
S. Madhan	20023245
I. Mohammed Shareef	20023248

ABSTRACT

The scope of the IEEE 802.21 (Media Independent Handover) standard is to develop a specification that provides link layer intelligence and other related network information to upper layers to optimize handovers between heterogeneous media. This includes links specified by 3GPP, 3GPP2 and both wired and wireless media in the IEEE 802 family of specifications. The IEEE 802.21 group defines the media independent handover function that will help mobile devices to roam across heterogeneous networks and stationary devices to switch over to any of the available heterogeneous networks around it.

The proposal can support handovers for both mobile and stationary users. For the mobile users handovers may occur due to a change in wireless link conditions. Alternatively, handovers may occur due to a gap in radio coverage as a result of terminal movement. For the stationary user handovers may become imminent when the environment around the user changes making one network more attractive than another. The user may choose an application which requires handover to a higher data rate channel, for example to download a large data file. Handovers should maximize service continuity, such as making a network transition during the pause in a voice call so as to minimize any perceptible interruption in service. The project aims at simulating such MIH event triggers in NS2 for inter 802.11 handover only. The QoS chosen are RSSI (Received Signal Strength) and number of dropped packets.

CONTENTS

CHAPTER	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	i
	ABSTRACT	ii
	LIST OF TABLES	v
	LIST OF FIGURES	vi
	LIST OF ABBREVIATIONS	vii
1	INTRODUCTION	1
	1.1 SCOPE AND NEED OF IEEE 802.21	1
	1.2 802.21 SPECIFICATIONS	2
	1.2.1 SERVICE CONTINUITY	2
	1.2.2 NETWORK SELECTION	3
	1.2.3 SECURITY	3
	1.2.4 POWER MANAGEMENT	3
	1.2.5 HANDOVERS DUE TO MOBILE TERMINAL MOVEMENT	3
	1.3 GENERAL MIH REFERENCE MODEL	4
	1.4 MEDIA INDEPENDENT EVENT SERVICE	5
	1.5 MEDIA INDEPENDENT COMMAND SERVICE	6
	1.5.1 MIH COMMANDS	7
	1.5.2 LINK COMMANDS	8
	1.6 MEDIA INDEPENDENT INFORMATION SERVICE	8
	1.7 INFORMATION SERVICE ELEMENTS	10
	1.8 LIST OF MIH FEATURES	10
2	NETWORK SIMULATOR 2	14
	2.1 BACKGROUND ON THE NS SIMULATOR	14
	2.2 AGENTS AND APPLICATIONS	14
	2.2.1 UDP AGENTS	14
	2.2.2 TCP AGENTS	15
	2.3 CONFIGURING A MOBILE NODE	16
	2.4 TRACE FILES	17

CONTENTS

CHAPTER	TITLE	PAGE NO.
	2.4.1 WIRED TRACE FORMATS	18
	2.4.2 WIRELESS TRACE FORMATS	19
2.5	NETWORK ANIMATOR	20
2.6	HIERARCHICAL ROUTING	21
	2.6.1 WIRED-CUM-WIRELESS SCENARIOS	23
2.7	802.11 IN NS2	24
3	STATIC AND DYNAMIC HANDOVER IN 802.11	28
3.1	PARAMETERS FOR IEEE 802.11 LINK LAYER QUALITY	28
3.2	STATIC HANDOVER	29
3.3	DYNAMIC HANDOVER	31
	3.3.1 NO. OF DROPPED PACKETS	31
	3.3.2 RECEIVED SIGNAL STRENGTH	34
4	HANDOVER USING EVENT TRIGGERS	35
4.1	SIMULATION MODEL	35
	4.1.1 LINK_GOING_DOWN EVENT TRIGGER	35
	4.1.2 LINK_ROLLBACK EVENT TRIGGER	36
	4.1.3 LINK_DOWN EVENT TRIGGER	36
4.2	SIMULATION SCENARIO	37
	4.2.1 QoS: RSSI	37
	4.2.2 PSEUDO CODE FOR GENERATION OF TRIGGERS WITH RSSI AS QoS	38
	4.2.3 QoS: NUMBER OF DROPPED PACKETS	39
	4.2.4 PSEUDO CODE FOR GENERATION OF TRIGGERS WITH NUMBER OF DROPPED PACKETS AS QoS	40
5	RESULTS AND DISCUSSION	41
5.1	OVERVIEW OF RESULTS	41
5.2	RECOMMENDED PARAMETER VALUES	42
5.3	HANDOVER LATENCY	42
5.4	SCOPE FOR FUTURE WORK	43
5.5	CONCLUSION	44
6	REFERENCES	45

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
1.1	MIH COMMANDS	7
1.2	LINK COMMANDS	8
1.3	LIST OF MIH FEATURES	10
2.1	WIRED TRACE FORMATS 1	18
2.2	WIRED TRACE FORMATS 2	18
2.3	WIRELESS TRACE FORMATS	19
5.1	RECOMMENDED RSSI THRESHOLD VALUES	42

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	GENERAL MIH REFERENCE MODEL	4
1.2	KEY MEDIA INDEPENDENT HANDOVER SERVICES	4
1.3	LINK EVENTS AND MIH EVENTS	6
1.4	INFORMATION SERVICE	9
2.1	NAM WINDOW	21
2.2	MAC SUPPORT IN NS2	24
2.3	MAC TRANSMISSIONS	25
3.1	STATIC HANDOVER	29
3.2	OUR SIMULATION SCENARIO	31
3.3	BEFORE HANDOVER	32
3.4	AFTER HANDOVER	32
3.5	NUMBER OF FORWARDED PACKETS WITHOUT HANDOVER	33
3.6	NUMBER OF FORWARDED PACKETS WITH HANDOVER	34
3.7	POWER DROPPING BELOW THRESHOLD	34
4.1	RSSI COMPARISON FOR 100 PACKETS	37
4.2	LINK_DOWN TRIGGER	37
4.3	LINK_GOING_DOWN TRIGGER	39
5.1	RECEIVED THRESHOLD VS DROP PACKETS	41
5.2	NUMBER OF LINK_GOING_DOWN TRIGGERS VS HANDOVER LATENCY	42
5.3	NUMBER OF DROPPED PACKETS VS HANDOVER LATENCY	43

LIST OF ABBREVIATIONS

S.NO	ACRONYM	DESCRIPTION
1.	AP	ACCESS POINT
2.	BS	BASE STATION
3.	GNI	GENERAL NETWORK INFORMATION
4.	HLI	HIGHER LAYER INFORMATION
5.	L1	LAYER 1, PHYSICAL LAYER (PHY)
6.	L2	LAYER 2, MEDIUM ACCESS CONTROL (MAC)
7.	L2.5	LAYER 2.5
8.	L3	LAYER 3
9.	LLI	LINK LAYER INFORMATION
10.	LAN	LOCAL AREA NETWORK
11.	MICS	MEDIA INDEPENDENT COMMAND SERVICES
12.	MIES	MEDIA INDEPENDENT EVENT SERVICE
13.	MIIS	MEDIA INDEPENDENT INFORMATION SERVICE
14.	MIH	MEDA INDEPENDENT HANDOVER
15.	MIP	MOBILE IP
16.	MN	MOBILE NODE
17.	PoA	POINTS OF ATTACHMENT
18.	WLAN	WIRELESS LOCAL AREA NETWORK

CHAPTER 1

INTRODUCTION

1.1 SCOPE AND NEED OF IEEE 802.21

The scope of the IEEE 802.21 (Media Independent Handover) standard is to develop a specification that provides link layer intelligence and other related network information to upper layers to optimize handovers between heterogeneous media. This includes links specified by 3GPP, 3GPP2 and both wired and wireless media in the IEEE 802 family of specifications.

The proposal can support handovers for both mobile and stationary users. For the mobile users handovers may occur due to a change in wireless link conditions. Alternatively, handovers may occur due to a gap in radio coverage as a result of terminal movement. For the stationary user handovers may become imminent when the environment around the user changes making one network more attractive than another. The user may choose an application which requires handover to a higher data rate channel, for example to download a large data file. Handovers should maximize service continuity, such as making a network transition during the pause in a voice call so as to minimize any perceptible interruption in service.

The IEEE802.21 standard supports cooperative use of both mobile terminals and network infrastructure. The mobile terminal is well-placed to detect available networks, and the infrastructure is in a position to store overall network information, such as neighborhood cell lists and the location of mobile devices. In general, both the terminals and the network point of attachments such as base stations or access points can be multimode, i.e. supporting different radio standards, and in some cases being capable of transmission on more than one interface simultaneously.

The network can have both micro cells (IEEE 802.11 or IEEE 802.15 coverage) and macro cells (3GPP, 3GPP2 or IEEE 802.16) and these will in general intersect. The handover process is typically based on measurements and triggers supplied from link layers on the terminal. These measurements may include signal quality measurements, synchronization time

differences, transmission error rates, etc. and are some of the metrics used in handover algorithms.

The IEEE 802.21 framework facilitates the network discovery and selection process by exchanging network information that helps mobile devices determine which networks are in their current neighborhoods. This network information could include information about the link type, the link identifier, link availability and link quality etc. of nearby network links. This process of network discovery and selection allows a mobile to connect to the most appropriate network based on certain mobile policies.

As the mobile moves between different network Points of Attachment (PoA), it is essential to maintain proper security associations between the communicating end-points. These security associations can be obtained both via lower layer and higher layer mechanisms.

1.2 802.21 SPECIFICATIONS

1.2.1 SERVICE CONTINUITY

Handovers may occur either between two different access networks or between two different points of attachment of a single access network. In such cases Service continuity is defined as the continuation of the service during and after the handover while minimizing aspects such as data loss and break time during the handover without requiring any user intervention. The change of access network may or may not be noticeable to the end user, but there should be no need for the user to re-establish the service. There may be a change in service quality as a consequence of the transition between different networks due to the varying capabilities and characteristics of the access networks. For example if the QoS supported by new access network is unacceptable, higher layer entities may decide not to handover or may terminate the current session after the handover based on applicable policies. This specification specifies essential elements which enable service continuity.

1.2.2 NETWORK SELECTION

Network selection is the continuous process of selecting the most appropriate network for any user operation at any given time. The selection can be based on various criteria such as required QoS, cost, user preferences, policies, etc. If the selected network is not the currently used network, then a handover to the preferred network may be required. The 802.21 standard

may specify the means for such information to be made available to the upper layers to enable effective network selection.

1.2.3 SECURITY

Events, commands and information messages carried between a MT (Mobile Terminal) and a network PoA (Point of Attachment) cannot be secured until the MT is securely associated with the network PoA. This association can be achieved either via lower or higher layers security mechanisms. Once such a secure association has been established between the MT and the network PoA, any messages exchanged between two MIH Function entities should retain integrity and be replay protected over a secure transport. Otherwise the exchanged MIH messages are prone to integrity, replay and man-in-the-middle attacks. The 802.21 standard may specify the means for security information to be made available to the upper layers to setup secure connections.

1.2.4 POWER MANAGEMENT

This specification provides for information that helps to preserve battery life. For example efficient 'sleep modes' can be managed based on real time link status, efficient scanning is achieved using neighbor maps of different networks and readily available reports of optimum link layer parameters.

1.2.5 HANDOVERS DUE TO MOBILE TERMINAL MOVEMENT

Handovers due to the mobile station speeds relative to the base station or access point are facilitated by providing real time link conditions and timely information about overlay micro-cells and macro-cells.

1.3 GENERAL MIH REFERENCE MODEL

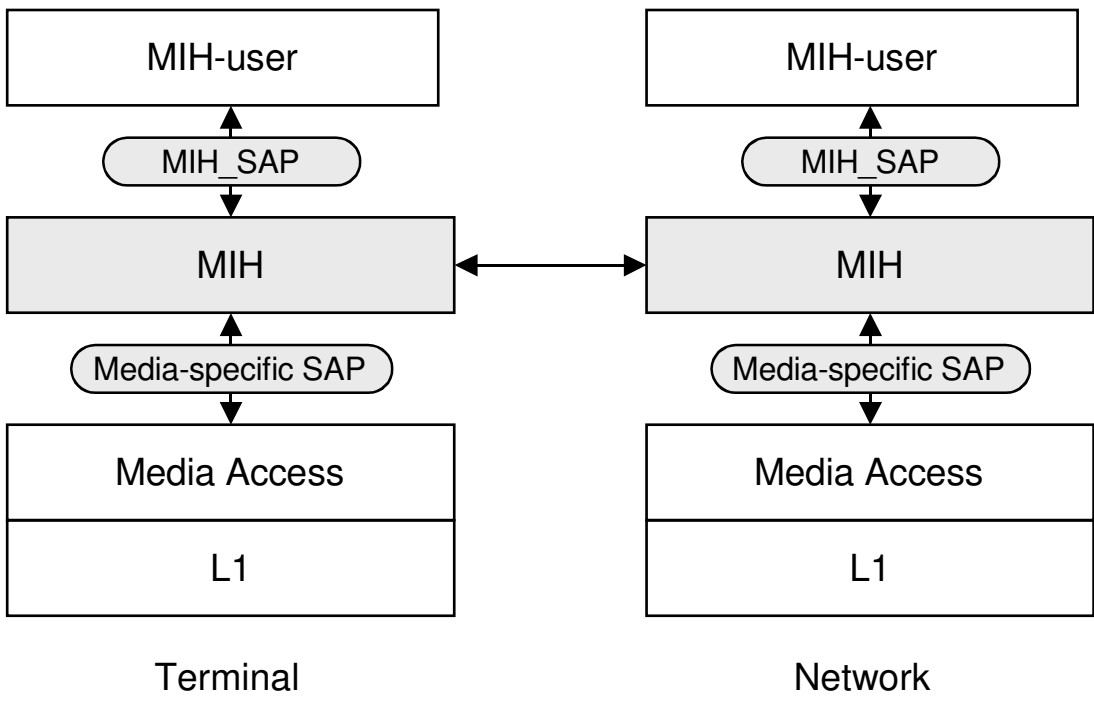


Figure 1.1 General MIH Reference model

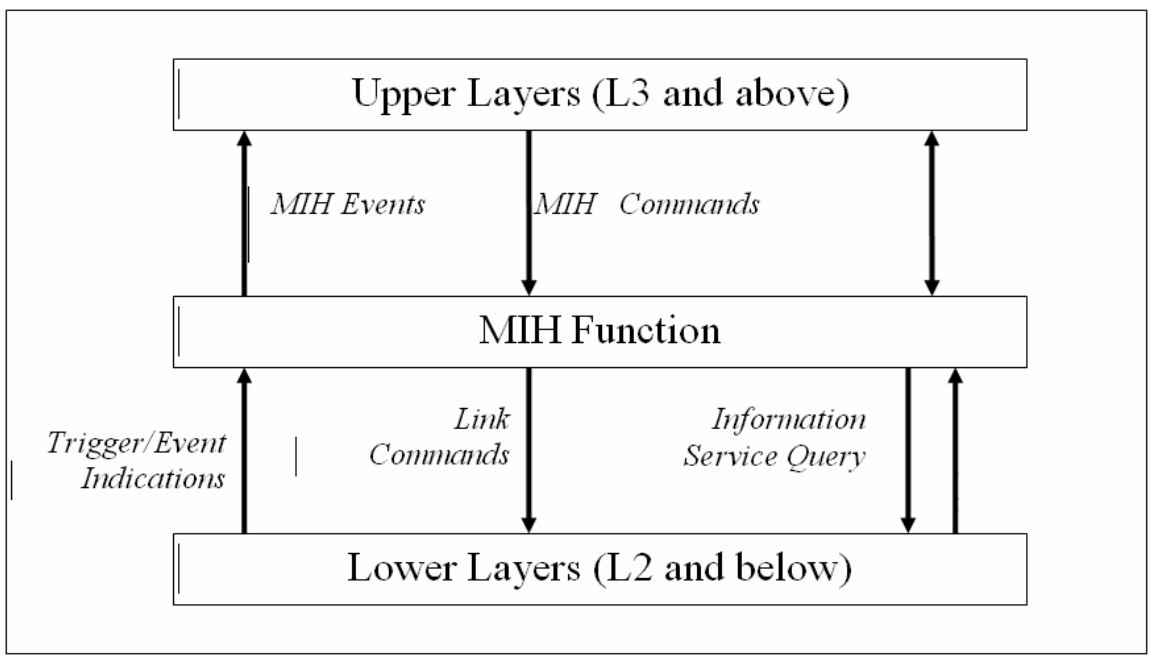


Figure 1.2 Key Media Independent Handover Services

1.4 MEDIA INDEPENDENT EVENT SERVICE

In general handovers can be initiated either by the mobile terminal or by the network. Events that can initiate handover may originate from MAC, PHY or MIH Function either at the mobile node or at the network point of attachment. This could be due to user or terminal mobility, state change in the environment or because of some management function on part of the network. Thus the source of these events can be either local or remote. A transport protocol is needed for supporting remote events. Security is another important consideration in such transport protocols.

Multiple higher layer entities may be interested in these events at the same time. Thus these events may need to have multiple destinations. Higher layer entities can register to receive event notifications from a particular event source. The MIH Function can help in dispatching these events to multiple destinations.

These events are treated as discrete events. As such there is no general event state machine. However in certain cases a particular event may have state information associated with it, such as the Link_Going_Down event discussed in this chapter. In such cases the event may be assigned an *identifier* and other related events may be associated with the corresponding event using this identifier.

From the recipient's perspective these events are mostly "advisory" in nature and not "mandatory". Layer 3 and above entities may also need to deal with reliability and robustness issues associated with these events. Higher layer protocols and other entities may prefer to take a more "defensive" approach when events originate remotely as opposed to when they originate locally.

The Event Service may be broadly divided into two categories, Link Events and MIH Events. Both Link and MIH Events typically traverse from a lower to higher layer. Link Events are defined as events that originate from event source entities below the MIH Function and typically terminate at the MIH Function. Entities generating Link Events include but is not restricted to various IEEE802-defined, 3GPP-defined and 3GPP2-defined interfaces. Within the MIH Function, Link Events may be further propagated, with or without additional processing, to upper layer entities that have registered for the specific event. Events that are propagated by the MIH to the upper layers is defined as MIH Events.

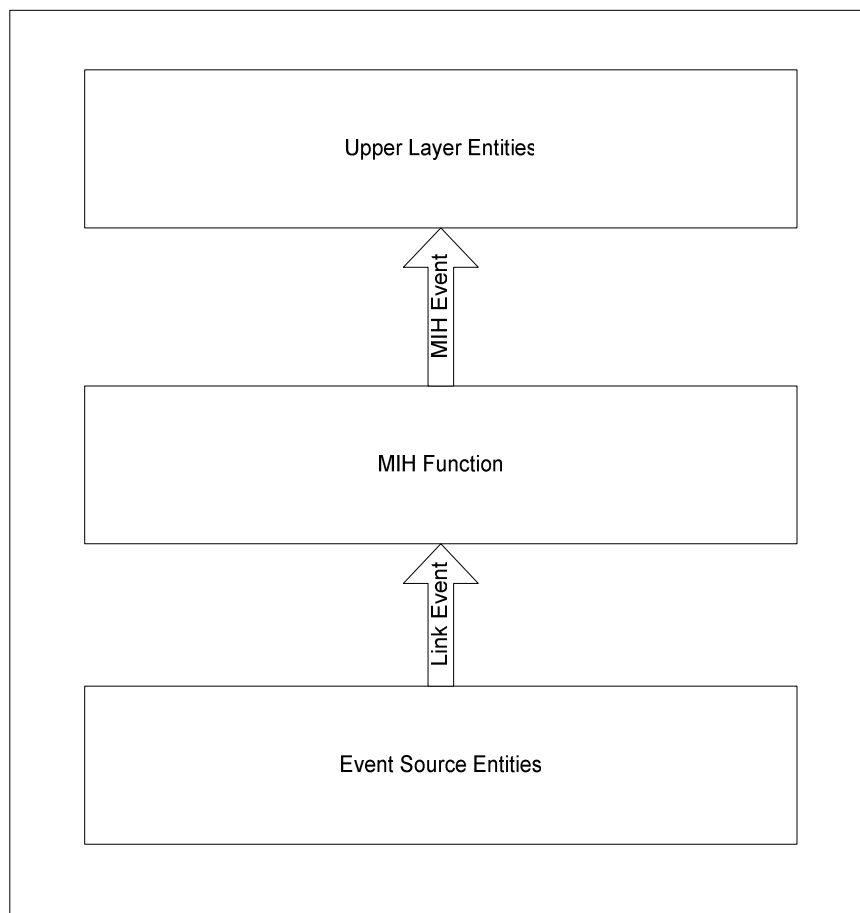


Figure 1.3 Link Events and MIH Events

1.5 MEDIA INDEPENDENT COMMAND SERVICE

The command service refers to the commands sent from the higher layers to the lower layers in the reference model. Upper layer MIH users may utilize command services to determine the status of links and/or control the multi-mode device for optimal performance. Command services may also enable MIH users to facilitate optimal handover policies. For example, the network may initiate and control handovers to balance the load of two different access networks.

The link status varies with time and terminal mobility. Information provided by MICS is dynamic information comprising of link parameters such as signal strength, link speed, etc, whereas information provided by MIIS is less dynamic or static in nature and is comprised of parameters such as network operators, higher layer service information, etc. MICS and MIIS Information could be used in combination by the terminal/network to facilitate the handover. For example, MIIS may use the information scanned by MICS to update the neighbor graphs.

A number of commands have been added to allow the upper layers to configure, control, and get information from the lower layers. A set of the commands that could be supported from upper layer to MIH should be defined in this specification. A set of command services that are provided by the L2 data link (MAC, functions such as MAC, Radio Resource Management etc. depending upon the L2 access link technology) and PHY should be defined in this specification.

1.5.1 MIH COMMANDS

MIH Command includes the commands from upper layer to MIH (e.g. upper layer mobility protocol to MIH, or policy engine to MIH, etc).

No	MIH Command	Local, Remote	Media Types	Parameters	Comments
1	MIH Poll	L, R	All		Poll the status of links
2	MIH Switch	L, R	All		Switch session between links
3	MIH Configure	L, R	All		Configure a link
4	MIH Scan	L, R	All		Scan a link

Table 1.1 MIH Commands

1.5.2 LINK COMMANDS

Link Command includes the commands from MIH to lower layer (e.g. MIH to MAC, or MIH to PHY). These commands mainly control the behavior of lower layer entities.

No	Link Command	Local, Remote	Media Types	Comments
1	LinkPowerUp	L	All	Power Up a link
2	LinkPowerDown	L	All	Power down a link

3	LinkConfigure	L	All	Configure a specific interface
4	LinkConnect	L	All	Connect on a specific link
5	LinkDisconnect	L	All	Disconnect the connection on specified link
6	LinkSleep	L	All	Put link into sleep mode
7	LinkScan	L	All	Scan the link for network PoA
8	LinkPoll	L	All	Poll a specific link

Table 1.2 Link Commands

1.6 MEDIA INDEPENDENT INFORMATION SERVICE

Media Independent Information Service (MIIS) provides a framework by which a MIHF (Media Independent Handover Function) entity, either in a station or in the network, can discover and obtain homogeneous or heterogeneous network information existing within a geographical area to facilitate the handovers. In the larger scope, the macro objective is to acquire a global view of the heterogeneous networks to facilitate seamless handover when roaming across these networks.

The important component of MIIS is Information Elements (IEs). Information Elements provide necessary information that is essential for a handover module to make intelligent handover decision. The list of supported Information Elements can be very large and may vary from one application to another. The MIIS provides support for only those Information Elements that are necessary for mobility applications. Figure 8-1 gives a high level description of scenarios that distinguish between two different types of mobility.

- Horizontal handover: A horizontal handover is made by switching between different Points of Attachment of the same access network. For example, switching between different APs within the 802.11 network (e.g. Inter ESS or inter subnet).
- Vertical handover: A vertical handover means a handover from one access network to another access network. For example switching between different PoAs across heterogeneous technologies (e.g. from WLAN to GPRS).

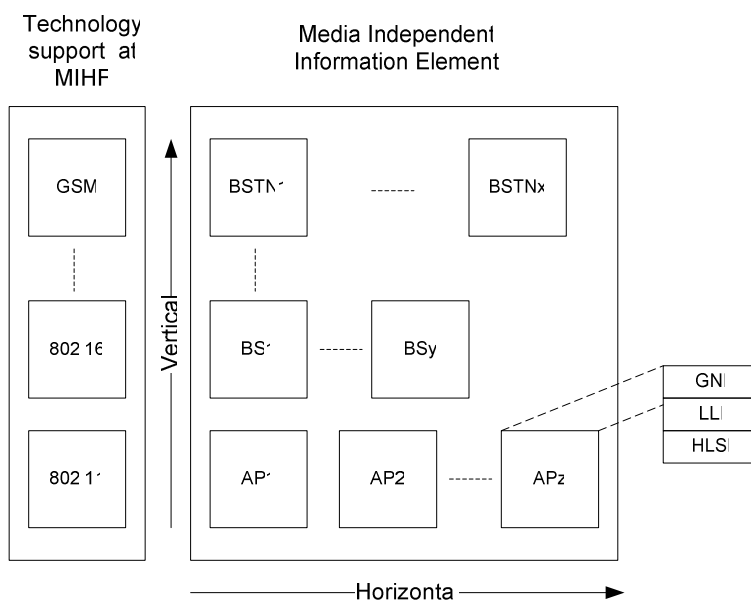


Figure 1.4 Information Service

1.7 INFORMATION SERVICE ELEMENTS

The information service elements can be classified into three groups:

1. General Network Information (GNI): These information elements give a general overview of the network, like location, name, network ID, POA (Point of Attachment) of the network, IP version, operator of the network, and so forth
2. Link Layer Information (LLI): These information elements include the information related to link layer layers such as, link layer parameters (channel, frequency, PHY types), data rates, neighbor information, security, QoS, and so forth.
3. Higher Layer Information (HLI): These information elements include higher layer services or applications that are supported by the respective network. Some examples are support for Multimedia Message Service (MMS), Mobile IP (MIP), Virtual Private Network (VPN), types of applications supported (e.g. VoIP, e-mail, IPsec VPNs, streaming media, location based), pricing of access (e.g. "a fee is be required" versus "access to the network is free"), use of NAT, roaming partners, and so forth

1.8 LIST OF MIH FEATURES

Category	Function
Event Service	Link Event Register Link Event Deregister Link Detected Link UP Link Down Link Going Down Link Event Rollback
Command Service	MIH Poll MIH Handover Initiate
MIH Protocol	Event Registration Link Events Handover Initiate (request/response) Poll (request/response)

Table 1.3 List of MIH Features

Link Trigger Generation Model

The following triggers have been implemented on the MN in the 802.11 simulation model.

Link Detected

At the MAC layer, a Link Detected event is generated upon the reception of a beacon message originating from another AP than the current AP of the MN (passive mode). If a MN is operating in the active mode, then it reports the result of the probe phase to the handover module. A Link Detected event is then generated for each AP that has been found.

Link Up

A Link Up is generated upon the reception of an Association Response message with a status code indicating that the MN is accepted in the cell.

Link Down

A Link Down event is generated when the MAC of the MN is disconnected from the AP. This occurs for any of the following cases:

- N consecutive packets have arrived with errors. By default N is set to 5. Section **Error! Reference source not found.** gives the effects of varying N on the handover performance.
- An Association Response message is received indicating that the MN is rejected from its current AP (i.e., status code field is different from “0” or unsuccessful).
- The BSSID has expired. The BSSID is advertised only in the Beacon message. By default, the BSSID expires if the MN does not receive a Beacon message during an interval greater than 3 times the Beacon interval, which is by default 3 x 100ms. Section **Error! Reference source not found.** determines the effects of different BSSID timeout intervals on the handover latency.
- The MN MAC is requested to connect to one AP. This decision can be either local or remote and leads to the generation of a link Down event for the current AP.

Link Going Down

A link Going Down is generated when the power level between two consecutive packets at the receiver is decreasing. Let P_n (in Watt) be the power level of the nth packet received, and P_{Th} be the power level threshold required for receiving packets without errors, a Link Going Down is triggered, if the following two conditions hold true:

$$P_n < \alpha P_{Th} \quad (1)$$

$$P_n < P_{n-1} \quad (2)$$

where α is a tuning parameter. Note that P_{Th} depends on the noise level of the operating environment and vendor fact sheets describing the receiver performance (for example, BER as a function of E_b/N_o). In the following, α will be called power level threshold coefficient.

Link Rollback

A Link Rollback is tightly coupled with a Link Going Down event. If a packet with higher power level is received immediately following a Link Going Down event, then the MAC layer generates a Link Rollback event to cancel the last link Going Down event generated. Thus, a Link Rollback event is generated if the following three conditions hold true:

$$P_{n-2} > P_{n-1} \quad (1)$$

$$P_{n-1} < \alpha P_{Th} \quad (2)$$

$$P_n > P_{n-1} \quad (3)$$

Link Handover Imminent

A Link Handover Imminent event is generated at the MAC layer when changing AP.

Link Handover Complete

A link Handover Complete is generated upon the reception of an Association Response message that indicates that the association with the target AP is accepted (i.e., status code field is set to “0” for successful association).

CHAPTER 2

NETWORK SIMULATOR 2

2.1 BACKGROUND ON THE NS SIMULATOR

The ns simulator covers a very large number of applications, of protocols, of network types, of network elements and of traffic models. We call these “simulated objects”. *ns* is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. The simulator supports a class hierarchy in C++, and a similar class hierarchy within the OTcl interpreter. The root of this hierarchy is the class TclObject. Users create new simulator objects through the interpreter.

2.2 AGENTS AND APPLICATIONS

In order to set up traffic flow between nodes, we must create agents and applications. The two main application used are the CBR (Constant Bit Rate) and the FTP (File Transfer Protocol) application. The Internet protocol used by FTP is TCP (Transport Control Protocol) and the one used by CBR is UDP (User Datagram Protocol).

2.2.1 UDP AGENTS

UDP agents are implemented in `udp.{cc, h}`. A UDP agent accepts data in variable size chunks from an application, and segments the data if needed. UDP packets also contain a monotonically increasing sequence number and an RTP timestamp. Although real UDP packets do not contain sequence numbers or timestamps, this sequence number does not incur any simulated overhead, and can be useful for tracefile analysis or for simulating UDP-based applications. The default maximum segment size (MSS) for UDP agents is 1000 byte:

```
Agent/UDP set packetSize_ 1000 ;# max segment size
```

The following commands are used to setup UDP agents in simulation scripts:

```
set udp0 [new Agent/UDP]
```

This creates an instance of the UDP agent.

```
$ns_ attach-agent <node> <agent>
```

This is a common command used to attach any <agent> to a given <node>.

```
$traffic-gen attach-agent <agent>
```

This is a class Application/Traffic/<traffictype> method which connects the traffic generator to the given <agent>.

For example, if we want to setup a CBR traffic flow for the udp agent, udp1, the following commands are given:

```
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$ns_ connect <src-agent> <dst-agent>
```

This command sets up an end-to-end connection between two agents (at the transport layer).

2.2.2 TCP AGENTS

TCP is a dynamic reliable congestion control protocol. It uses acknowledgements created by the destination to know whether packets are well received; lost packets are interpreted as congestion signals. TCP thus requires bidirectional links in order for the acknowledgements to return to the source. There are a number of variants of the TCP Protocol. Running an TCP simulation requires creating and configuring the agent, attaching an application-level data source (a traffic generator), and starting the agent and the traffic generator.

There are two major types of TCP agents: one-way agents and a two-way agent. One-way agents are further subdivided into a set of TCP senders (which obey different congestion and error control techniques) and receivers (“sinks”). The two-way agent is symmetric in the sense that it represents both a sender and receiver. It is still under development.

```
set src [new Agent/TCP/FullTcp] ;           # create agent
set sink [new Agent/TCP/FullTcp] ;         # create agent
$ns_ attach-agent $node_(s1) $src ;        # bind src to node
$ns_ attach-agent $node_(k1) $sink ;       ;# bind sink to node
$src set fid_ 0 ;                          # set flow ID field
```

```

$sink set fid_ 0 ;                               # set flow ID field
$ns_ connect $src $sink ;                         # active connection src to sink
$ns_ at $start-time "$ftp start" ;               # start ftp flow

```

2.3 CONFIGURING A MOBILE NODE

A mobilenode consists of network components like Link Layer (LL), Interface Queue (IfQ), MAC layer, the wireless channel nodes transmit and receive signals from etc. At the beginning of a wireless simulation, we need to define the type for each of these network components. Additionally, we need to define other parameters like the type of antenna, the radio-propagation model, the type of ad-hoc routing protocol used by mobilenodes etc.

First, we need to configure nodes before we can create them. Node configuration API may consist of defining the type of addressing (flat/hierarchical etc), the type of adhoc routing protocol, Link Layer, MAC layer, IfQ etc. The configuration API can be defined as follows:

```

                                     (parameter examples)
# $ns_ node-config -addressingType flat or hierarchical or expanded
#                   -adhocRouting   DSDV or DSR or TORA
#                   -llType         LL
#                   -macType         Mac/802_11
#                   -propType        "Propagation/TwoRayGround"
#                   -ifqType         "Queue/DropTail/PriQueue"
#                   -ifqLen          50
#                   -phyType         "Phy/WirelessPhy"
#                   -antType         "Antenna/OmniAntenna"
#                   -channelType     "Channel/WirelessChannel"
#                   -topoInstance    $topo
#                   -energyModel     "EnergyModel"
#                   -initialEnergy   (in Joules)
#                   -rxPower         (in W)
#                   -txPower         (in W)
#                   -agentTrace      ON or OFF
#                   -routerTrace     ON or OFF
#                   -macTrace        ON or OFF
#                   -movementTrace   ON or OFF

```

All default values for these options are NULL except:

addressingType: flat

2.4 TRACE FILES

There are a number of ways of collecting output or trace data on a simulation. Generally, trace data is either displayed directly during execution of the simulation, or (more commonly) stored in a file to be post-processed and analyzed. There are two primary but distinct types of monitoring capabilities currently supported by the simulator.

The first, called *traces*, record each individual packet as it arrives, departs, or is dropped at a link or queue. Trace objects are configured into a simulation as nodes in the network topology, usually with a Tcl “Channel” object hooked to them, representing the destination of collected data (typically a trace file in the current directory). The other types of objects, called *monitors*, record counts of various interesting quantities such as packet and byte arrivals, departures, etc. Monitors can monitor counts associated with all packets, or on a per-flow basis using a *flow monitor*.

2.4.1 WIRED TRACE FORMATS

The trace starts with one of four possible characters.

Event	Abbreviation	Type	Value
Normal Event	r: Receive	%g %d %d %s %d %s %d %d.%d %d.%d %d %d	
	d: Drop	double	Time
	e: Error	int	Source Node
	+: Enqueue	int	Destination Node
	-: Dequeue	string	Packet Name
		int	Packet Size
		string	Flags
		int	Flow ID
		int	Source Address
		int	Destination Address
		int	Sequence Number

		int	Unique Packet ID
--	--	-----	------------------

Table 2.1 Wired Trace formats 1

Depending on the packet type, the trace may log additional information:

Event	Type	Value
TCP Trace	%d 0x%x %d %d	
	int	Ack Number
	hexadecimal	Flags
	int	Header Length
	int	Socket Address Length
Satellite Trace	%.2f %.2f %.2f %.2f	
	double	Source Latitude
	double	Source Longitude
	double	Destination Latitude
	double	Destination Longitude

Table 2.2 Wired Trace 2

2.4.2 WIRELESS TRACE FORMATS

The wireless traces begin with one of four characters. This is followed by flag/value pairs. The first letter of flags with two letters designates the flag type:

- N: Node Property
- I: IP Level Packet Information
- H: Next Hop Information
- M: MAC Level Packet Information
- P: Packet Specific Information

Event	Abbreviation	Flag	Type	Value
Wireless	s: Send	-t	double	Time (* For Global Setting)

Event	r: Receive d: Drop f: Forward	-Ni	int	Node ID
		-Nx	double	Node X Coordinate
		-Ny	double	Node Y Coordinate
		-Nz	double	Node Z Coordinate
		-Ne	double	Node Energy Level
		-NI	string	Network trace Level (AGT, RTR, MAC, etc.)
		-Nw	string	Drop Reason
		-Hs	int	Hop source node ID
		-Hd	int	Hop destination Node ID, -1, -2
		-Ma	hexadecimal	Duration
		-Ms	hexadecimal	Source Ethernet Address
		-Md	hexadecimal	Destination Ethernet Address
		-Mt	hexadecimal	Ethernet Type
		-P	string	Packet Type (arp, dsr, imep, tora, etc.)
-Pn	string	Packet Type (cbr, tcp)		

Table 2.3 Wireless Trace formats

An example TCP trace

```

+ 0.94176 2 3 tcp 1000 ----- 0 0.0 3.0 25 40
+ 0.94276 2 3 tcp 1000 ----- 0 0.0 3.0 26 41
d 0.94276 2 3 tcp 1000 ----- 0 0.0 3.0 26 41
+ 0.95072 2 0 ack 40 ----- 0 3.0 0.0 14 29
- 0.95072 2 0 ack 40 ----- 0 3.0 0.0 14 29
    
```

```
- 0.95176 2 3 tcp 1000 ----- 0 0.0 3.0 21 36  
+ 0.95176 2 3 tcp 1000 ----- 0 0.0 3.0 27 42
```

2.5 NETWORK ANIMATOR

Nam is a Tcl/TK based animation tool for viewing network simulation traces and real world packet tracedata. The design theory behind nam was to create an animator that is able to read large animation data sets and be extensible enough so that it could be used indifferent network visualization situations. Under this constraint nam was designed to read simple animation event commands from a large trace file. In order to handle large animation data sets a minimum amount of information is kept in memory. Event commands are kept in the file and reread from the file whenever necessary.

The first step to use nam is to produce the trace file. Usually, the trace file is generated by ns. During an ns simulation, user can produce topology configurations, layout information, and packet traces using tracing events in ns. However any application can generate a nam trace file. When the trace file is generated, it is ready to be animated by nam. Upon startup, nam will read the tracefile, create topology, pop up a window, do layout if necessary, and then pause at time 0. Through its user interface, nam provides control over many aspects of animation. These functionalities will be described in detail in the USER INTERFACE section.

You can either start nam with the command 'nam <nam-file>' where '<nam-file>' is the name of a nam trace file that was generated by ns, or you can execute it directly out of the Tcl simulation script for the simulation which you want to visualize.

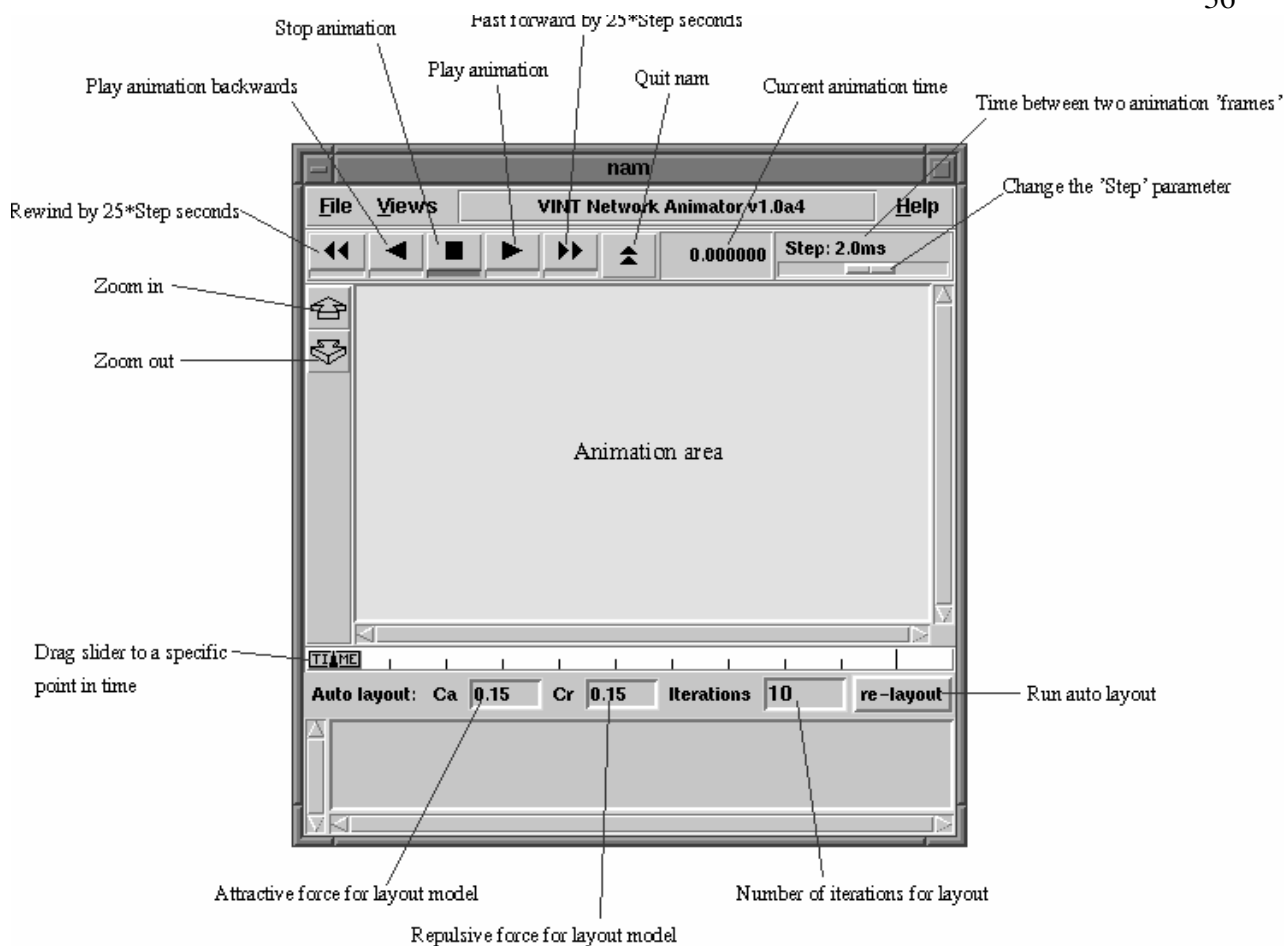


Figure 2.1 NAM Window

2.6 HIERARCHICAL ROUTING

Hierarchical routing was mainly devised, among other things, to reduce memory requirements of simulations over very large topologies. A topology is broken down into several layers of hierarchy, thus downsizing the routing table. The table size is reduced from n^2 , for flat routing, to about $\log n$ for hierarchical routing. However some overhead costs results as number of hierarchy levels are increased. Optimum results were found for 3 levels of hierarchy and the current ns implementation supports up to a maximum of 3 levels of hierarchical routing. To be able to use hierarchical routing for the simulations, we need to define the hierarchy of the topology as well as provide the nodes with hierarchical addressing.

In flat routing, every node knows about every other node in the topology, thus resulting in routing table size to the order of n^2 . For hierarchical routing, each node knows only about those nodes in its level. For all other destinations outside its level it forwards the packets to the border router of its level. Thus the routing table size gets downsized to the order of about $\log n$.

Hierarchical routing requires some additional features and mechanisms for the simulation. For example, a new node object called *HierNode* is been defined for hier routing. Therefore the user must specify hierarchical routing requirements before creating topology.

```
$ns set-address-format hierarchical
```

This sets the node address space to a 3 level hierarchy assigning 8 bits in each level.

Class AddrParams is used to store the topology hierarchy like number of levels of hierarchy, number of areas in each level like number of domains, number of clusters and number of nodes in each cluster.

```
AddrParams set domain_num_ 2
lappend cluster_num 2 2
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 2 3 2 3
AddrParams set nodes_num_ $eilastlevel
```

This defines a topology with 2 domains, say D1 and D2 with 2 clusters each (C11 & C12 in D1 and C21 & C22 in D2). Then number of nodes in each of these 4 clusters is specified as 2, 3, 2 and 3 respectively. The default values used by AddrParams provide a topology with a single domain with 4 clusters, with each cluster consisting of 5 nodes.

2.6.1 WIRED-CUM-WIRELESS SCENARIOS

The main problem facing the wired-cum-wireless scenario was the issue of routing. In ns, routing information is generated based on the connectivity of the topology, i.e how nodes are connected to one another through Links. Mobilenodes on the other hand have no concept of links. They route packets among themselves, within the wireless topology, using their routing protocol. So how would packets be exchanged between these two types of nodes? So a node

called BaseStationNode is created which plays the role of a gateway for the wired and wireless domains. The BaseStationNode is essentially a hybrid between a Hierarchical node1 (HierNode) and a MobileNode. The basestation node is responsible for delivering packets into and out of the wireless domain. In order to achieve this we need Hierarchical routing.

Each wireless domain along with its base-station would have a unique domain address assigned to them. All packets destined to a wireless node would reach the base-station attached to the domain of that wireless node, who would eventually hand the packet over to the destination (mobilenode). And mobilenodes route packets, destined to outside their (wireless) domain, to their base-station node. The base-station knows how to forward these packets towards the (wired) destination.

The DSDV agent on having to forward a packet checks to see if the destination is outside its (wireless) subnet. If so, it tries to forward the packet to its base-station node. In case no route to base-station is found the packet is dropped. Otherwise the packet is forwarded to the next hop towards the base-station, which is then routed towards the wired network by base-station's classifiers.

The DSR agent, on receiving a packet destined outside its subnet, sends out a route-query for its base-station in case the route to base-station is not known. The data packet is temporarily cached while it waits to hear route replies from base-station. On getting a reply the packet is provided with routing information in its header and sent away towards the base-station. The base-station address de-muxes routes it correctly toward the wired network.

2.7 802.11 IN NS2

The various MAC layers inherently supported by NS2 is shown in the diagram below.

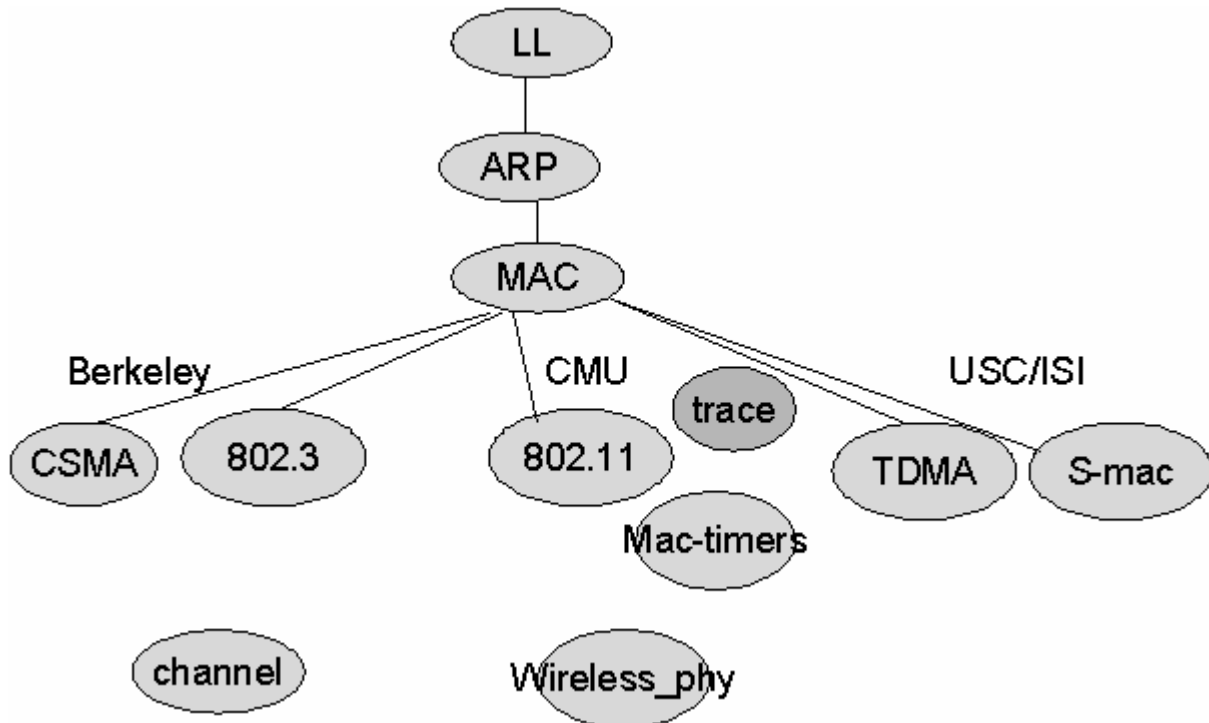


Figure 2.2 MAC Support in NS2

Local Variables:

- pktTx_
- pktRx_
- Macstate_ :
- index_ : mac address

Basic functions of General MAC class.

- recv (packet, handler)
 - This is the entry from upper target (a callback handler is given as a parameter) to send a packet to MAC. After the MAC transmit this packet successfully, it will use this callback handler to inform the upper target that MAC is idle and give another packet if there are packets buffered.
- SendUp
 - entry for receiving a packet. Sendup is function is directly called by the lower target of MAC, might be "netif" or "phy". And this function directly calls the upper_target. Because the uplink to upper_target does not involve any physical transmission delay, it does not need any timer and state change here. The question is that when the MAC is in MAC_RECV state? The answer is: The

MAC here is supposed to be full-duplex and receive can be happened simultaneously. it does not care about collisions etc. This is a general MAC class

- SendDown
 - used to sending packet down. Called by `recv().init` a timer for tx, and the timer handler is defined to call `resume()`.
- Handler* `callback_;`
 - when MAC is idle, the upper target has to be callback.
- Resume()
 - When tx timer out, reset MAC as idle state and callback.
- Discard
 - When a packet has to be drop, the `drop_ (NsObject*)` of bi-connector class has to be called to handle this, usually `drop (p, why)` is used. Why is a string of drop reason, in `cmu-trace.h`. three-charecter string is defined to describe those reasons in the trace file, such as "BSY", "CBK"....

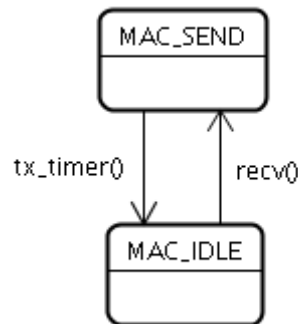


Figure 2.3 MAC Transmissions

TRANSMITTING A PACKET

Roughly takes the following path (when no errors or congestion):

`recv()` -> `send()` -> `sendDATA()` and `sendRTS()` -> start defer timer

-> `deferHandler()` -> `check_pktRTS()` -> `transmit()`

-> `recv()` -> receive timer started

-> recv_timer() -> recvCTS() -> tx_resume() -> start defer timer -> rx_resume()

-> deferHandler() -> check_pktTx() -> transmit()

-> recv() -> receive timer started

-> recv_timer() -> recvACK() -> tx_resume() -> callback_ -> rx_resume() -> done!

When the first RTS fails:

recv() -> send() -> sendData() and sendRTS() -> start defer timer

-> deferHandler() -> check_pktRTS() -> transmit -> start send timer

-> send_timer() -> RetransmitRTS() -> tx_resume() -> backoff timer started

backoffHandler() -> check_pktRTS() -> transmit

Rest is the same as above.

RECEIVING A PACKET

Roughly takes the following path (when no errors or congestion):

recv() -> receive timer started

-> recv_timer() -> recvRTS() -> sendCTS() -> tx_resume() -> start defer timer -> rx_resume()

-> deferHandler() -> check_pktCTRL() -> transmit()

-> recv() -> receive timer started

-> recv_timer() -> recvDATA() -> sendACK() -> tx_resume() -> start defer timer -> uptarget_-
>recv()

-> deferHandler() -> check_pktCTRL() -> transmit() -> start send timer

-> send_timer() -> tx_resume() <- done.

CHAPTER 3

STATIC AND DYNAMIC HANDOVER IN 802.11

3.1 PARAMETERS FOR IEEE 802.11 LINK LAYER QUALITY

We propose a list of parameters to characterize the quality of the IEEE 802.11 link layer and that can constitute a basis to compute the Link Down and Link Going Down events. These parameters can be used in both simulation modeling and real implementations.

- **Data rate:** This represents the theoretical bit rate that an interface is able to operate on. Since the coverage area is generally larger for lower data rates, devices may adapt their transmission rates according to the received signal strength. If the AP is capable of supporting different bit rates, then it might change its data rate with a particular station in order to maintain the association with this station. Therefore, the change of the data rate used between an AP and a MN might be an indication that the MN is getting far from the AP.
- **RSSI (Received Signal Strength Indicator):** The RSSI measures the signal strength of the received frames. This parameter is a function of the distance between the MN and its AP and can be used to detect that a link is going down. However the RSSI also depends on the environment, interference, noise, channel propagation properties, antenna design. A drop of the RSSI does not necessarily mean that the MN is about to leave its AP's cell, but it can be due to temporary interference for example.
- **Packet error threshold or number of packets with errors:** When a MN is losing its association with its AP, the number of packets received with errors increases. Therefore, the number of packet with errors might be a criterion to determine that the link is going down.
- **Missed beacon threshold or number of missed beacons:** When a MN is out of range of its current AP, it can not receive the beacon messages that are periodically sent by the AP. A MN is able to determine the number of beacon messages that it has missed because the beacon interval is included in each beacon. Therefore, the number of consecutive beacon messages missed by a MN might be a criterion to determine that a link is down.
- **Number of retransmissions:** If the number of retransmission needed to successfully send a frame to the AP is increasing, it means that either the data frame is lost or the

acknowledgment is lost. When the MN is getting close to the border of its AP's cell, more errors can be introduced in packets and therefore more retransmissions are incurred.

- **Number of duplicate frames:** If the MN is receiving multiple instances of the same data frame, it means that the acknowledgment packets are lost. Therefore, it might indicate that the AP is not receiving correctly the MN's frames and might indicate that the link is going down.

3.2 STATIC HANDOVER

In this scenario we have generated a static handover scenario based on the rate required by the mobile applications. First we assume that an application requires a rate of 0.1Mbps. Then another application starts at the mobile node which requires a rate of 0.25 Mbps. The first base station is not able to support a rate of 0.25 Mbps that is required by the application. Hence the control of the mobile node is handed over to another base station that is able to support the required rate.

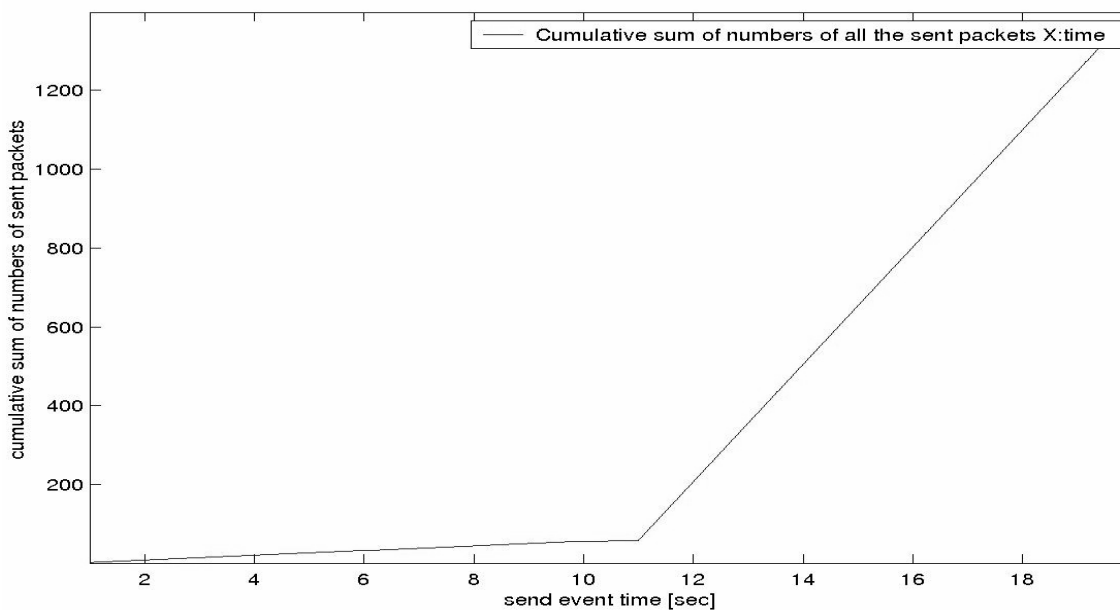


Figure 3.1 Static Handover

From the graph we can see that no packet is received during the handover time (Between 10.5 and 11.5 s). This is evident from the flat portion of the graph.

We have used hierarchical routing as our scenario includes both wired and wireless nodes. There are two wired nodes connected together by duplex links. There are two base stations and a mobile node. The TCP packets are generated by wired node 0 and the destination

node is the mobile node. The packets are routed through the wired node1 and one of the base stations. The routing information for wired nodes are based on connectivity of the topology, i.e. how nodes are connected to one another through Links. This connectivity information is used to populate the forwarding tables in each wired node. However wireless nodes have no concept of "links". Packets are routed in a wireless topology using their adhoc routing protocols which build forwarding tables by exchanging routing queries among its neighbours. So inorder to exchange pkts among these wired and wireless nodes, we use base-stations which act as gateways between the two domains. We segregate wired and wireless nodes by placing them in different domains. Domains and sub-domains (or clusters as they are called here) are defined by means of hierarchical topology structure. The base station serves as an interface between the wired and the wireless nodes. We have put the wired nodes in two separate domains and the base stations in two separate domains. The mobile node is in the same domain as a base station but in a different cluster.

Number of domains = 3

Number of clusters in each domain = 2, 1, 1

Number of nodes in each cluster = 1, 1, 2, 1

At the start of simulation the mobile node 4 is associated with base station 2, i.e. it is in the second domain with address 1.0.1. Then at a later time we change this association to another domain with base station 3 and address 2.0.1. Our future work involves making this association Dynamic, i.e. during run-time based on QoS parameters like dropped packets.

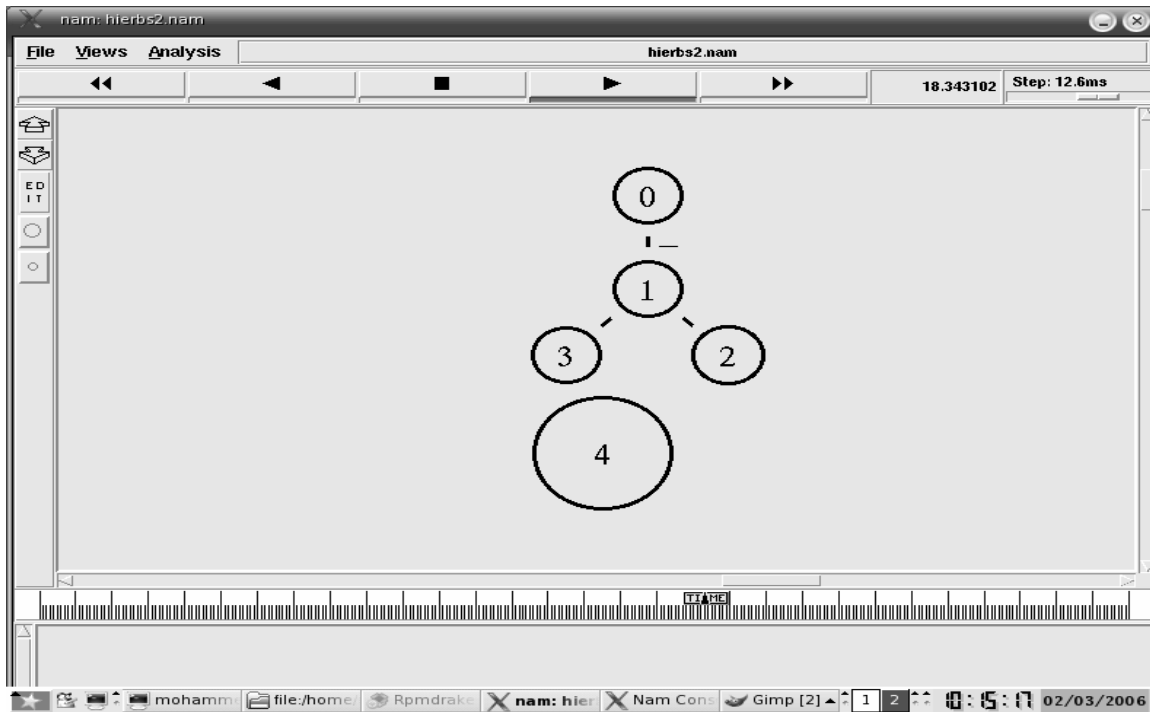


Figure 3.2 Our Simulation scenario

3.3 DYNAMIC HANDOVER

3.3.1 NO. OF DROPPED PACKETS

In this scenario we consider the number of dropped packets as our QoS. Some applications specify the maximum amount of packet loss that can be tolerated. We try to keep our packet threshold below this maximum packet loss allowed. When the numbers of dropped packets cross the specified threshold the association of the base station changes dynamically.

We have modified the file `cmu-trace.cc` and included a counter for the number of dropped packets. Every time a packet is dropped and written in the trace file the counter is increased. When the count exceeds the threshold dynamic handover occurs.

Node 0 – Wired node

Node 1 and 2 – Base station nodes

Node 3 – Mobile node

Node 3 is the source node and node 0 is the sink for the UDP packets. First the packets are routed through base station 1.

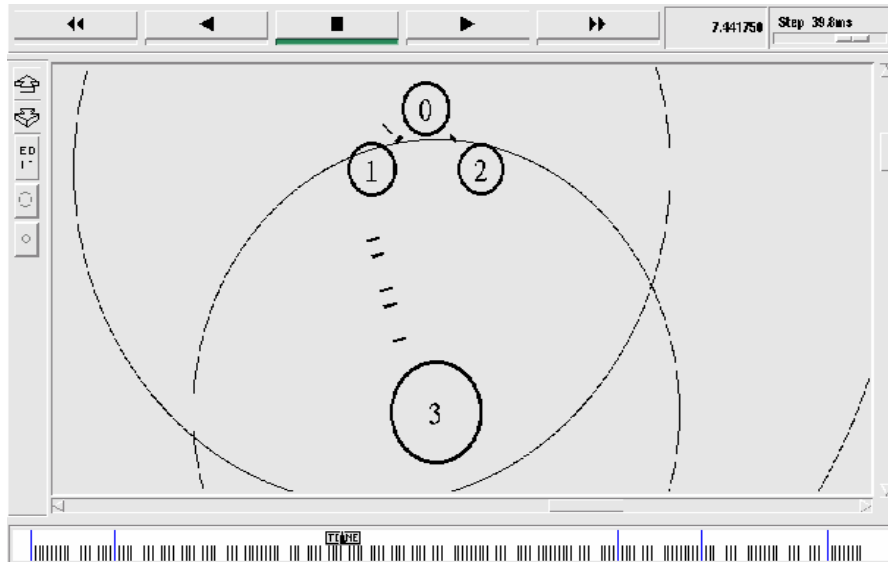


Figure 3.3 Before Handover

When the number of dropped packets crosses the threshold the association of the base station changes and the UDP packets are routed through base station 2.

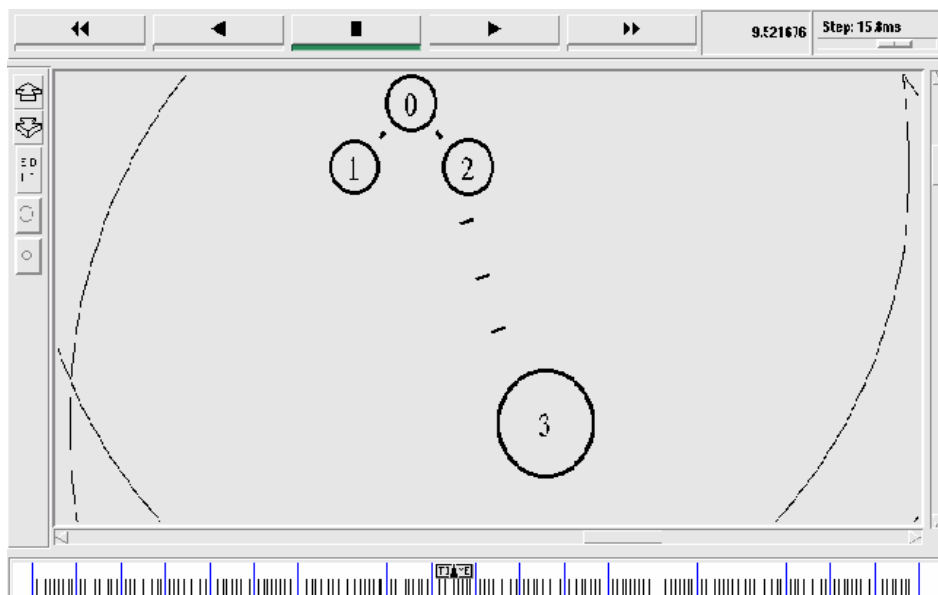


Figure 3.4 After Handover

Without handover we can see that base station 1 is only forwarding the packets between the source and the sink. After some time the number of dropped packets keep increasing till all the packets are dropped. Base station 2 is not involved in this case

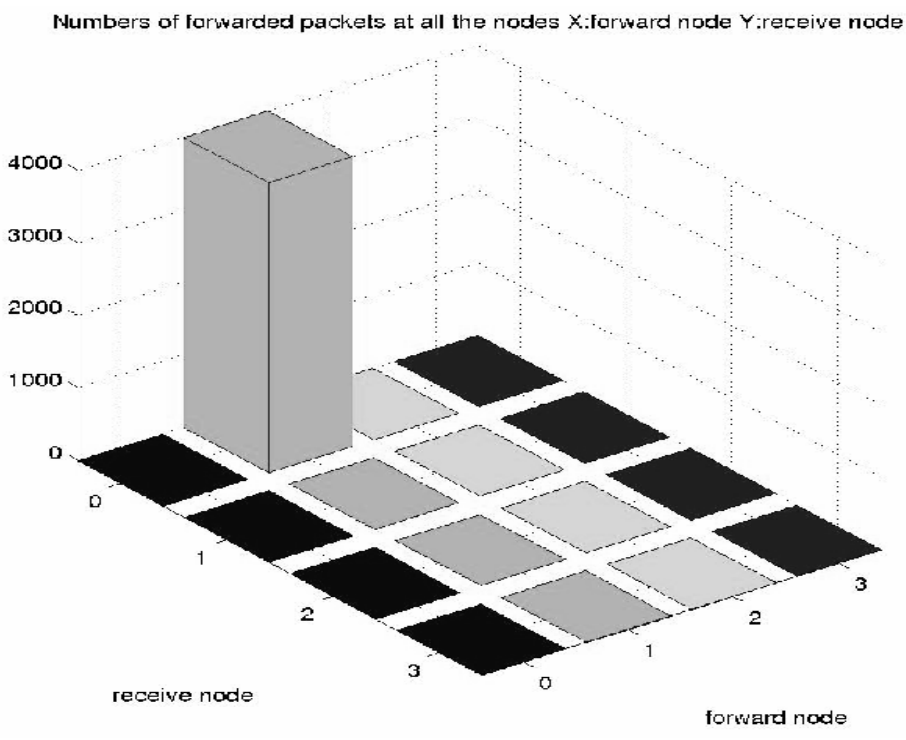


Figure 3.5 Number of forwarded packets without handover

In this graph we can see that both the base stations forward packets between the source and the sink. After the number of dropped packets cross the threshold, base station 2 starts forwarding the packets.

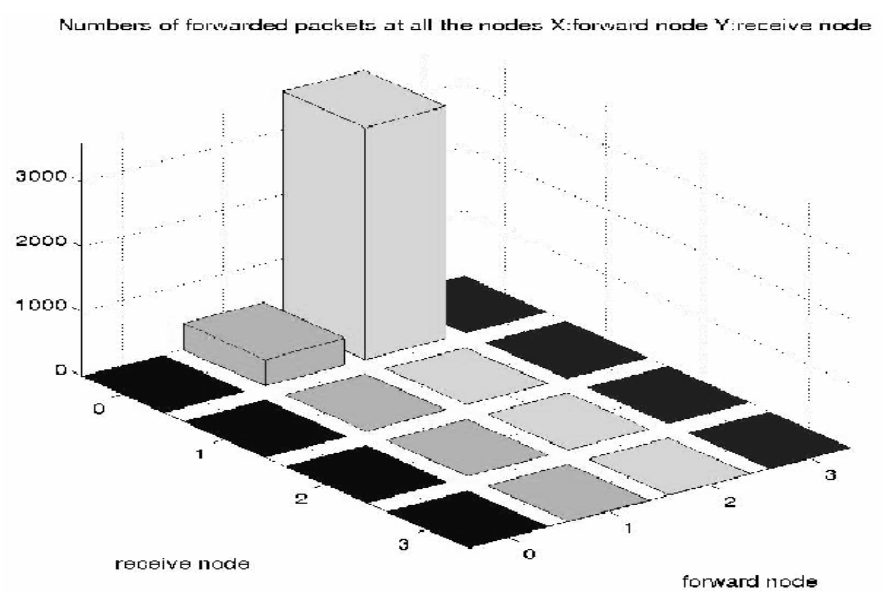


Figure 3.6 Number of forwarded packets with handover

3.3.2 RECEIVED SIGNAL STRENGTH

The second QoS that we have considered is the received signal strength. We have modified the file mac-802_11.cc to include the code for dynamic handover. We check the power of each packet received. If the received power goes below the specified threshold handover occurs dynamically.

```
.....  
.....  
RXThresh_ 7.45e-08  
Rx pkt power 7.44925e-08  
changing association ABRUPT H0 dynamically QoS power  
  
packet collision reached 1 due to low power  
.....  
.....  
.....
```

Figure 3.7 Power dropping below threshold

CHAPTER 4

HANDOVER USING EVENT TRIGGERS

4.1 SIMULATION MODEL

In this section, we describe the simulation model using NS-2 version 2.28.

Key modifications to NS-2

To support our simulations, different modifications and improvements to NS-2 have been necessary. The modifications include:

- Media Independent Handover agent implementing 802.21 events and commands
- LINK_GOING_DOWN EVENT TRIGGER
- LINK_ROLLBACK EVENT TRIGGER
- LINK_DOWN EVENT TRIGGER

4.1.1 LINK_GOING_DOWN EVENT TRIGGER

A link Going Down is generated when the power level between two consecutive packets at the receiver is decreasing. Let P_n (in Watt) be the power level of the n th packet received, and P_{Th} be the power level threshold required for receiving packets without errors, a Link Going Down is triggered, if the following two conditions hold true:

$$P_n < \alpha P_{Th} \quad (1)$$

$$P_n < P_{n-1} \quad (2)$$

where α is a tuning parameter. Note that P_{Th} depends on the noise level of the operating environment and vendor fact sheets describing the receiver performance (for example, BER as a function of E_b/N_o). In the following, α will be called power level threshold coefficient.

4.1.2 LINK_ROLLBACK EVENT TRIGGER

A Link Rollback is tightly coupled with a Link Going Down event. If a packet with higher power level is received immediately following a Link Going Down event, then the MAC layer generates a Link Rollback event to cancel the last link Going Down event generated. Thus, a Link Rollback event is generated if the following three conditions hold true:

$$P_{n-2} > P_{n-1} \quad (1)$$

$$P_{n-1} < \alpha P_{Th} \quad (2)$$

$$P_n > P_{n-1} \quad (3)$$

4.1.3 LINK_DOWN EVENT TRIGGER

A Link Down event is generated when the MAC of the MN is disconnected from the AP. This occurs for any of the following cases:

- N consecutive packets have arrived with errors. By default N is set to 5. Section **Error! Reference source not found.** gives the effects of varying N on the handover performance.
- An Association Response message is received indicating that the MN is rejected from its current AP (i.e., status code field is different from “0” or unsuccessful).
- The BSSID has expired. The BSSID is advertised only in the Beacon message. By default, the BSSID expires if the MN does not receive a Beacon message during an interval greater than 3 times the Beacon interval, which is by default 3 x 100ms. Section **Error! Reference source not found.** determines the effects of different BSSID timeout intervals on the handover latency.
- The MN MAC is requested to connect to one AP. This decision can be either local or remote and leads to the generation of a link Down event for the current AP.

4.2 SIMULATION SCENARIO

4.2.1 QoS: RSSI

In the scenario we considered earlier for received threshold power, the decrease in received power maybe temporary due to local fading characteristics. Handovers should not be initiated under such conditions. Unnecessary handovers might lead to ping pong effect in

which handover occurs continuously between two base stations. To avoid this, some sorts of triggers are needed to specify when the handover process. Such triggers are specified in the IEEE 802.21 proposal.

```

RXThresh_ 7.45e-08      rxthresh[99] 7.38843e-08
rxthresh[4] 7.38843e-08
rxthresh[3] 7.38843e-08
rxthresh[2] 7.38843e-08
rxthresh[1] 7.38843e-08
rxthresh[0] 7.38818e-08
RxPr       7.38818e-08
           LINK GOING DOWN

changing association dynamically (QoS power)

packet collision reached 98 due to low power

```

Figure 4.1 RSSI comparison for 100 packets

```

.....
.....
HO_INITIATE at time 6.2306 ..... power[1]
7.43826e-08
power[0] 7.43808e-08
RxPr     7.43808e-08
        LINK DOWN

changing association dynamically (QoS power)

No. of low power packets reached 27

HO_COMPLETE at time 6.25292 .....
.....
.....

```

Figure 4.2 Link_Down trigger

Whenever a packet is received with power level lesser than the previous packet a counter 'x' is incremented and the LINK_GOING_DOWN trigger is generated. When the power of the next packet is higher than the previous packet power, another counter is incremented. This may happen due to rapid fluctuations in the link. To avoid this three successive packets must be received with higher power levels to reset the counter 'x' thereby canceling the previous LINK_GOING_DOWN trigger and generate a LINK_ROLLBACK trigger. When the counter 'x' reaches a threshold value of 25 the LINK_DOWN trigger is generated and the handover is initiated.

4.2.2 PSEUDO CODE FOR GENERATION OF TRIGGERS WITH RSSI AS QoS

The pseudo code for generation of triggers with RSSI as QoS is as follows:

```

BEGIN
SET A THRESHOLD FOR RSSI
CHECK EVERY RECEIVED PACKET FOR POWER LEVEL
  If (P (n) < RSSI THRESHOLD) then
    If (POWER (n) < POWER (n-1)) then
      GENERATE LINK_GOING_DOWN TRIGGER AND INCREMENT
      LINK_GOING_DOWN COUNTER
    Else
      INCREMENT LINK_ROLLBACK COUNTER
      If (LINK_ROLLBACK COUNTER > 2) then
        CANCEL THE LAST LINK_GOING_DOWN TRIGGER BY
        RESETTING THE LINK_GOING_DOWN COUNTER
        RESET THE LINK_ROLLBACK COUNTER
      Endif
    Endif
  Endif
  If (LINK_GOING_DOWN COUNTER > THRESHOLD) then
    GENERATE LINK_DOWN TRIGGER
    INITIATE HANDOVER
    RESET LINK_ROLLBACK COUNTER
    RESET LINK_GOING_DOWN COUNTER
  Endif
Else
  RESET LINK_GOING_DOWN COUNTER
Endif

```

4.2.3 QoS: NUMBER OF DROPPED PACKETS

The events taking place on every received packet is monitored by the cmu-trace file. If a packet is dropped a corresponding counter is incremented and the Link_Going_Down trigger is generated. If the successive packets (more than 2) are not dropped, the previous Link_Going_Down trigger is cancelled and the drop counter is reset. When the


```
Endif
Else
INCREMENT LINK_ROLLBACK COUNTER
If (LINK_ROLLBACK COUNTER > 2) then
    CANCEL THE LAST LINK_GOING_DOWN TRIGGER BY
    RESETTING THE LINK_GOING_DOWN COUNTER
    RESET THE LINK_ROLLBACK COUNTER
Endif
Endif
```

CHAPTER 5

RESULTS AND DISCUSSION

5.1 OVERVIEW OF RESULTS

For proper handover, we must fix the threshold number of packets or the threshold power at an optimum value. A high threshold for the number of packets will increase the packet loss before handover and a low threshold for the number of packets will cause unnecessary handovers. Similarly if the power threshold is low this will lead to a temporary drop in connection before handover occurs. If the power threshold is high unnecessary handovers may occur. Hence we plot the receive threshold Vs the number of packets. We can see that the plot varies as the mobility of the node varies. Also it does not follow a well defined mathematical relationship. Hence, we try to find the optimum value for the various thresholds depending on the mobility.

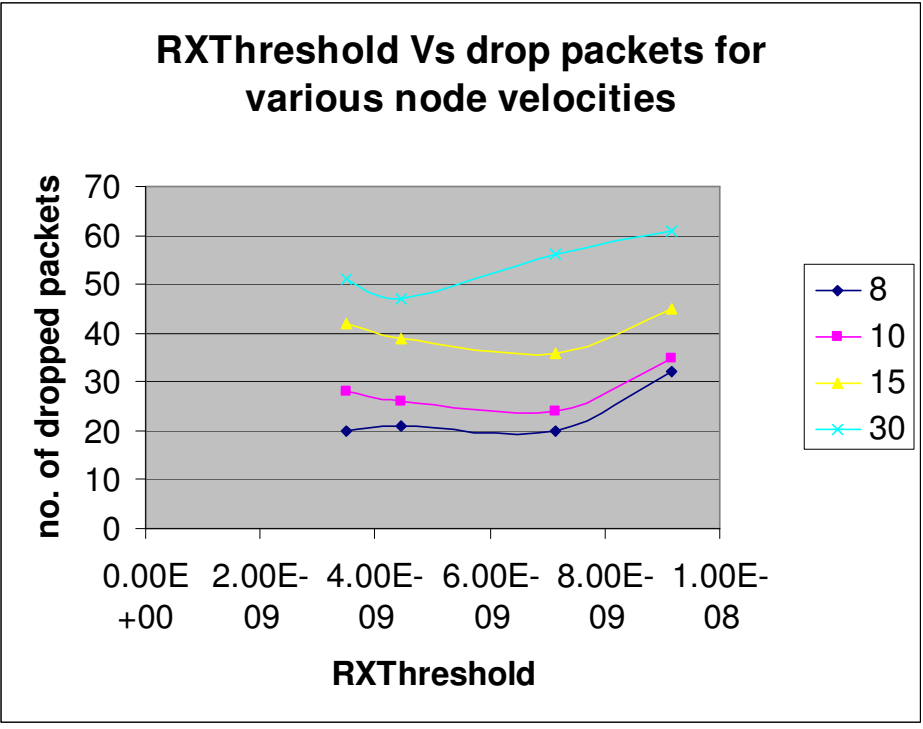


Figure 5.1 Received threshold Vs drop packets

5.2 RECOMMENDED PARAMETER VALUES

From the graph, we can see that the number of dropped packets increases as the node velocity increases. There is a threshold value for every value of the node velocity, at which the number of dropped packets is minimum. We can fix the value at which the number of dropped packets is minimum, as our threshold value.

QoS Parameter	Mobility (m/s)	Value (W)
RSSI threshold	8	7.15E-09
RSSI threshold	10	7.15E-09
RSSI threshold	15	7.15E-09
RSSI threshold	30	4.45E-09

Table 5.1 Recommended RSSI Threshold Values

5.3 HANDOVER LATENCY

Handover latency is the time interval between the first Link_Going_Down trigger generation and the HO_Complete generation.

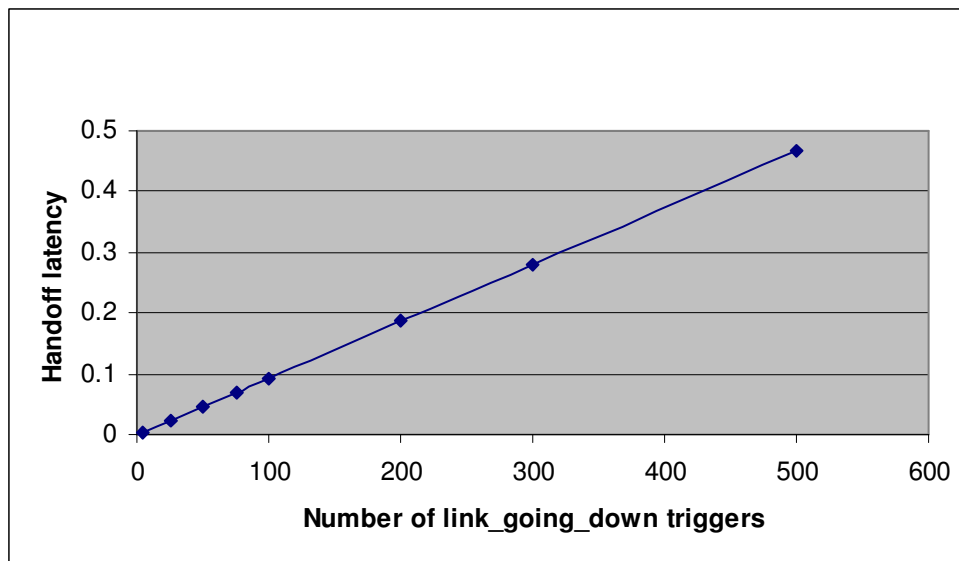


Figure 5.2 Number of link_going_down triggers Vs Handover latency

As we increase the number of Link_Going_Down triggers needed for generating the Link_Down trigger, the handover latency increases. This is evident from Figure 5.2. Hence, to decrease the handover latency a lower number of Link_Going_Down triggers must be considered. But if this value is set to be too low, then unnecessary handovers may occur.

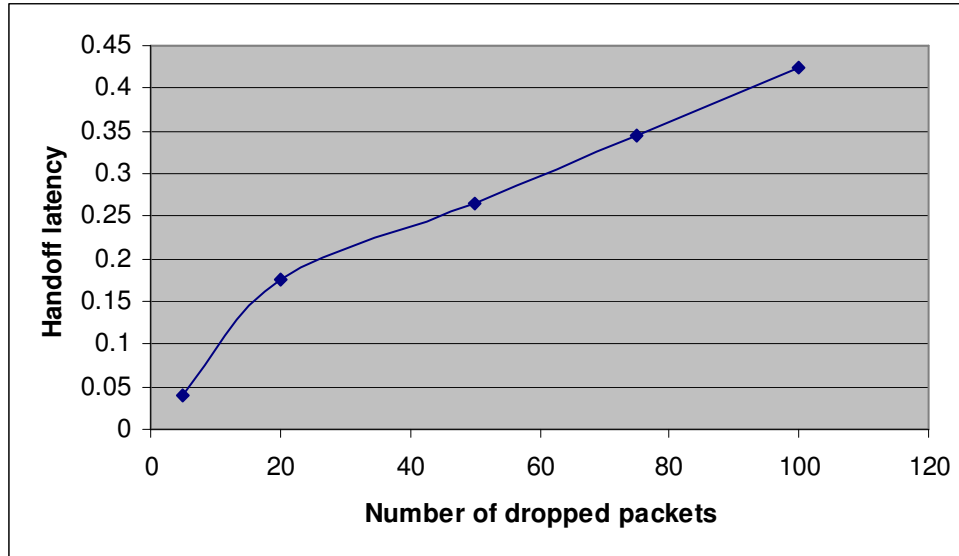


Figure 5.3 Number of dropped packets Vs Handover latency

The same can be inferred from Figure 5.3 where the plot is between the number of dropped packets and the handover latency.

5.4 SCOPE FOR FUTURE WORK

1. The 802.21 triggers have been generated for the 802.11 standard only as the handover is between two 802.11 APs. The future work may involve the implementation of multiple MAC support in a single wireless node thereby supporting heterogeneous handover. This will involve making changes to the node structure in NS2 and other related files so that a single wireless node can support multiple MACs and multiple PHYs.

2. There are various other QoS to be considered like missed beacons and number of packets received with error. In our simulations we have considered only the number of dropped packets and the RSSI as the QoS parameters.

3. Also, the authentication and the other security aspects when a mobile node registers with a base station can be considered.

5.5 CONCLUSION

This thesis presented a simulation using NS2 for the 802.21 triggers that are necessary for Medium Independent Handover. The simulation results have been used to find the threshold values of the QoS considered for various mobile nodes.

REFERENCES

1. Ashutosh Dutta, Subir Das, Telcordia technologies (2005), “Seamless Handover across Heterogeneous Networks”, IEEE 80.21 session proceedings.
2. Giuseppe Bianchi, Ilenia Tinnirello, Luca Scalia (2005), “Handover across heterogeneous wireless systems: a platform-independent control logic design”, IEEE 802.21 session proceedings.
3. Vivek Gupta, Xiayou (May 2005), “Draft Text for Media Independent Handover Specification”, Proposal for IEEE 802.21
4. John K. Ousterhout, “TCL and the Toolkit”, Wiley Publications