AN EFFICIENT SYSTEM TO DETECT AND BLOCK THE UNAUTHORIZED USAGE OF MOBILE PHONES IN RESTRICTED AREAS

A PROJECT REPORT

Submitted By

V.ALAMELU	**30609205003**
R.RAJESHWARI	**30609205075**
SHRIYA RAVIPRASAD	**30609205095**

in the partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

JEPPIAAR ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI - 600 025

APRIL 2013

ANNA UNIVERSITY

CHENNAI - 600 025

JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI ROAD, CHENNAI-119

BONAFIDE CERTIFICATE

This is to certify that this Project Report "**AN EFFICIENT SYSTEM TO DETECT AND BLOCK THE UNAUTHORIZED USAGE OF MOBILE PHONES IN RESTRICTED AREAS**" is the bonafide work of "**V.ALAMELU, R.RAJESHWARI** and **SHRIYA RAVIPRASAD**" who carried out the project under my supervision.

SUPERVISOR **HEAD OF THE DEPARTMENT**

Submitted for the examination held on ……………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We are very much indebted to the chairman, **Hon. Col. Dr. Jeppiaar M.A., B.L.,Ph.D,** the secretary Dr **P. Chinnadurai M.Phil., B.Ed., Ph.D,** **t**he principal **Dr. Sushil Lal das M.Sc(Engineering)., Ph.D.** and the PG principal **Dr. N. Thangavel, Ph.D.**

We wish to thank our HOD, **Dr. R. Sabitha,** our internal guide for guiding us to take up the project. I also thank our project coordinator for his enthusiastic support shown towards our project. Their efforts in establishing methodical work routine have lead to the successful completion of this project.

We are elated to place in record our sincere thanks and gratitude to **Mr.T.S Pradeepkumar, VIT, Chennai** who gave us the technical support to complete this project.

We wish to convey our sincere thanks to all the teaching and non-teaching Staff of The department of Information Technology, without whose co-operation this venture would not have been a success.

We wish to thank our parents, our friends and other well wishers who helped us to complete this work successfully.

ABSTRACT

The main objective of our project is to design a “Nova Intelligent Device”, which classifies the authorized and unauthorized numbers in a particular area and allows communication establishment to only the authorized numbers and blocks all unauthorized users. The key feature in this device is that, we are using a jammer, which will block all the signals from or to any base station to a mobile phone. Here, we make use of two Nova intelligent devices, where one is inside the jammed area, named as device1 and another is outside the jammed area, named as device2. There is already a separate channel allocated between our device2 and the base stations of different service providers in that area, with prior authorization from the respective service providers. For any incoming call to a person in the restricted area, the call would be first directed to our Device2 via the separate channel assigned. Then the Device2 would forward the call to Device1 which is inside the jammed area. For any outgoing call, the call would be first directed to our device1 and then to the device2 and then to the respective base station if in case the call is made by an authorized person.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	**ii**
	LIST OF FIGURES	**v**
1	**INTRODUCTION**	
	1.1 Motivation	1
	1.2 Objective	2
2	**SYSTEM ANALYSIS**	
	2.1 Existing System	3
	2.1.1 Drawback	4
	2.2 Proposed System	4
	2.3 Feasibility Study	15
	2.3.1 Economical Feasibility	15
	2.3.2 Operational Feasibility	17
	2.3.3 Technical Feasibility	19
3	**SYSTEM SPECIFICATION**	
	3.1 Hardware Requirements	20
	3.2 Software Requirements	20
4	**SOFTWARE DESCRIPTION**	
	4.1 Network Simulator (ns2)	21
	4.2 Features	30
5	**PROJECT DESCRIPTION**	
	5.1 Problem Definition	33
	5.2 Overview of the Project	33
	5.3 Module Description	34
	5.3.1 Modules	34
6	**SYSTEM IMPLEMENTATION**	
	6.1 Source Code	35

7	**SYSTEM TESTING**	
	7.1 Introduction	48
	7.2 Unit Testing	48
	7.3 Integration Testing	50
	7.4 System Testing	50
	7.5 Acceptance Testing	51
8	**CONCLUSION AND FUTURE ENHANCEMENTS**	52
	8.1 Future Enhancements	52
9	**Appendix**	
	9.1 Screen Shots	53
10	**References**	

LIST OF FIGURES

S.NO	TITLE	PAGE NO
2.1	Basic Architecture	6
2.2	Circuit Diagram of Device1	7
2.3	Circuit Diagram of Device2	8
2.4	Jammer	9
2.5	Frequency Mixer	12
2.6	Outgoing Call Concept	14
4.1	Basic Architecture of NS	23
4.2	Sample Network Topology	24
4.3	Two Language Architecture of NS2	26
4.4	Sample Chain of Events in Discrete Event Simulation	27
4.5	Node Architecture	30

CHAPTER 1
INTRODUCTION

Mobile phone usage has become indispensable in today's scenario. But people are making use of it even in restricted areas such as schools, colleges, factories etc. In such a situation, one might think of using a cell phone jammer to block its usage. But in this case, even if a person wants to make an important call, he would not be able to. Hence, we have come out with a solution to overcome this problem.

1.1 Motivation

The fact to be kept in our mind is that, mobile phones are very much essential and hence it must be allowed to bring to colleges. But then it must not be used during the class hours. Hence, we have proposed the system to ensure that it is not used illegally.

In this case when a call is made, it is forwarded to our "Nova Intelligent Device", which is already fed with a list of authorized numbers. This device will check if the call is made from an authorized user. If yes, then the call will be forwarded to the respective base station and then the communication would be established. If it's from a foreign number then the call will be denied. Same way, for an incoming call, the signal from the base station is directed to our device, which will check if it's for authorized person and establish communication if yes.

The major advantage of our project is that, a person will be allowed to carry mobile phone, but only in the restricted area he will be denied the service if he/she is unauthorized. When the person moves out of the area, he/she would be able to use the phone as normally as before.

1.2 Objective

The main objective of our project is to design a “Nova Intelligent Device”, which classifies the authorized and unauthorized numbers in a particular area and allows communication establishment to only the authorized numbers and blocks all unauthorized users. The key feature in this device is that, we are using a jammer, which will block all the signals from or to any base station to a mobile phone. Here, we make use of two Nova intelligent devices, where one is inside the jammed area, named as device1 and another is outside the jammed area, named as device2. Another point to be noted here is that, there is already a separate channel allocated between our device2 and the base stations of different service providers in that area, with prior authorization from the respective service providers. For any incoming call to a person in the restricted area, the call would be first directed to our Device2 via the separate channel assigned. Then the Device2 would forward the call to Device1 which is inside the jammed area. For any outgoing call, the call would be first directed to our device1 and then to the device2 and then to the respective base station if in case the call is made by an authorized person.

CHAPTER 2
SYSTEM ANALYSIS

2.1 Existing System

There are many related systems which are being developed for the mobile phone usage, detection and restriction. One such device is a novel mobile detector which detects the presence of GSM signal and indicates it. Here, when any mobile is used in the prohibited place, this Device will detect that mobile signal through the antenna. In that particular place, when a mobile signal is received, the receiver in this device will receive the signal through the antenna. When mobile receive the signal at the particular place, the alarm makes the sound for indication of the mobile and one LED will glow for the indication. Then with this device GSM Module is attached to send the Short Message service (SMS) to the registered number in the microcontroller. This detector is used to detect the presence of mobile. When it detects any mobile, it gives a signal to the PIC16F877A microcontroller. The controller when receives this signal will turn ON the buzzer circuit and will also send the detected message to some particular mobile numbers via the GSM module. Also the information is displayed in the LCD module as “MOBILE DETECTED”.

The Jammer is another device which deals with blocking of mobile usage irrespective of the user. Mobile phone jammers are radio communications transmitters designed to interfere with licensed services operated by mobile carriers. A mobile phone works by communicating with its service network via a base station. A mobile phone jammer typically works by preventing the mobile phone from receiving signals from base stations. As a result, the mobile phone does not attempt to transmit to a base station, even though it may be within range. A cell phone jammer emits signals in the same frequency range that cell phones use, effectively blocking their transmissions by creating strong interference. Someone using a cell phone within the range of a jammer will lose signal, but have no way of knowing a jammer was the reason. The phone will simply indicate poor reception strength. The major disadvantage here is that mobile usage will be completely blocked and a person will not be able to make a call during the emergency situation.

2.1.1 Drawbacks

The novel mobile detector detects the presence of a mobile in a particular area. But just its detection, serves no use, because, people can keep on using the mobile phone.

If a jammer is used, then no one will be able to use the phone itself. If any important call has to be made by an important person, they will not be able to. Hence, our system is addressing these drawbacks.

2.2Proposed System

Mobile phones have become indispensable in today’s environment, but at the same time mobile phone usage has to be restricted in certain places such as schools, colleges, hospitals, factories, places of worship etc; But blocking the mobile phone usage completely by using a jammer is not the right solution because higher officials in schools, colleges and factories need to communicate with other people officially. Hence, in such a scenario our system proves to be very much efficient and necessary since it is capable implementing selective blocking of mobile phone usage in restricted areas.

Our system aims at detecting and selectively blocking the usage of cell phones in restricted areas. The various works related to cell phone usage restriction restricts the usage completely irrespective of the user. But our system makes use of two special devices namely **nova intelligent device1 and nova intelligent device2**. Nova intelligent device1 is placed within the restricted area whereas the nova intelligent device2 is placed outside the restricted area. So, when a call is made in the restricted area, the call will be directed to our device1, which then checks whether the number is authorized, if in case the number is authorized then the call will be directed to the respective base station through our device2 which is placed outside the restricted area. When the detected number is found to be unauthorized the call will not be directed to the base station and hence the call is blocked. Thus, with the help of our system the authorized users can make a call or receive a call without any interruption whereas the unauthorized users will be denied the access.

Our nova intelligent device1 consists of a mobile jammer which sends out signals in the range 800-1900Mhz.It also acts like a miniature base station. It consists of a transceiver and a receiver which sends out signals in a particular frequency range which has been bought by the organization from the Federal Communication Commission(FCC).Say, for example, the frequency range bought the organization is around 200-220Mhz then the transceiver in the base station would send out signals in the range 200-220Mhz.Our device also consists of an EEPROM(Electrically Erasable Programmable Read Only Memory) into which the authorized numbers are stored so when a call is directed to our device1 it checks in the EEPROM whether the number is authorized or not.

Our nova intelligent device2 is placed outside the restricted area. Device1 is connected with Device2 via a separate communication channel with a different frequency, say 500MHZ. All the communication from the Device1 will always be first forwarded to Device2. Device2 has separate communication channel links with different Base stations of various Service providers in that area. This channel can be with the usual frequency used by the Base Stations.

Say, for example, let us consider that our proposed system is being implemented in a college. So, in this case the restricted area is a college, in a college the authorized users would be the Chairman, Principal, HODs and the staffs. The numbers of these authorized users would be stored in the EEPROM.

When a call is made the call would be directed to our nova intelligent device1 which then checks in the EEPROM whether the number is authorized or not. The advantage of using an EEPROM is that when a new number has to be added to the list of authorized users or when a number is to be deleted from a list of authorized users it can be easily done in an EEPROM since it's electrically erasable.

The architecture diagram of our proposed system being implemented in a college campus would be as follows:

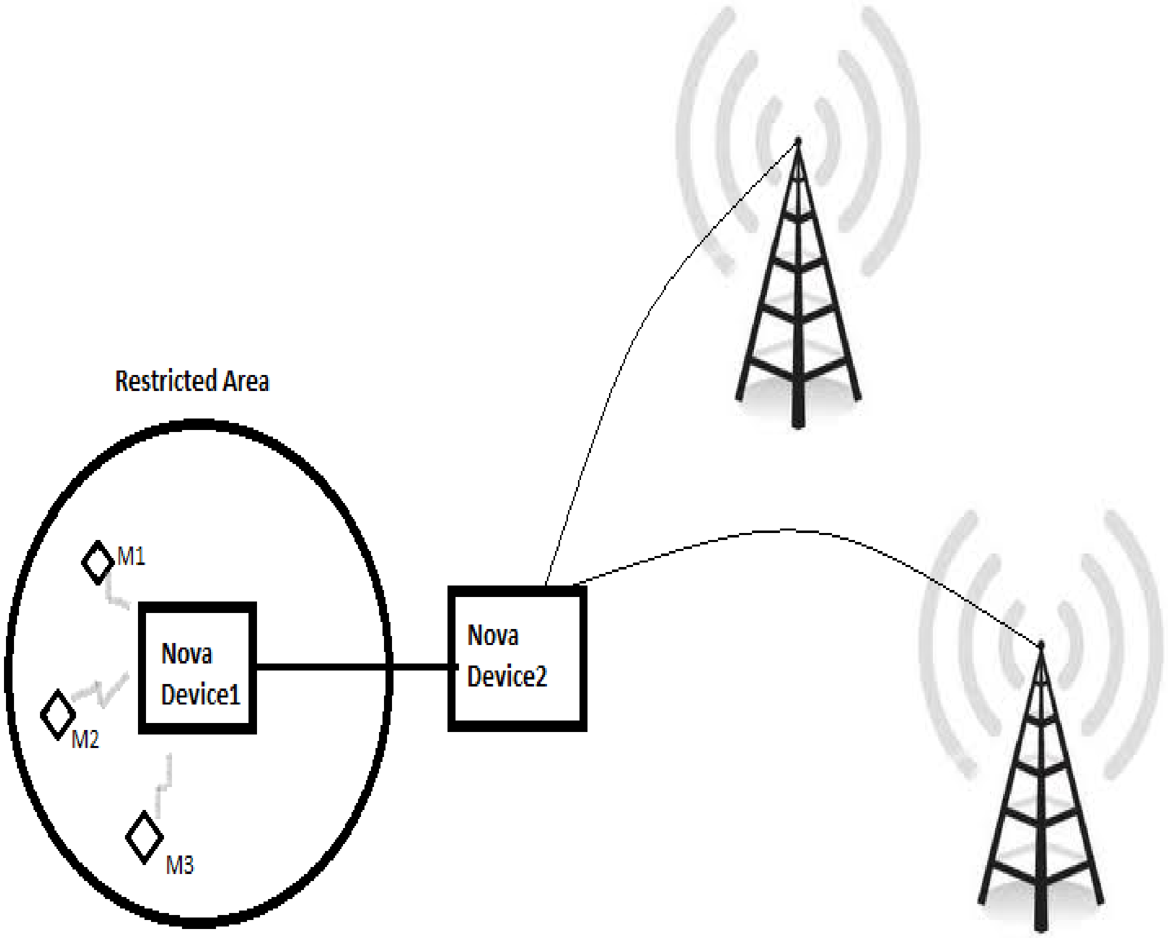


Figure 2.1: Basic Architecture

2.2.1 NOVA INTELLIGENT DEVICE1

Our nova intelligent device1 which is placed inside the restricted area has the following parts:

- a mobile jammer
- an EEPROM(Electrically Erasable Programmable Read Only Memory)
- a processor
- a GSM/RF(transceiver and receiver)
- a switch and a controller.

The circuit diagram of our nova intelligent device1 is as follows:

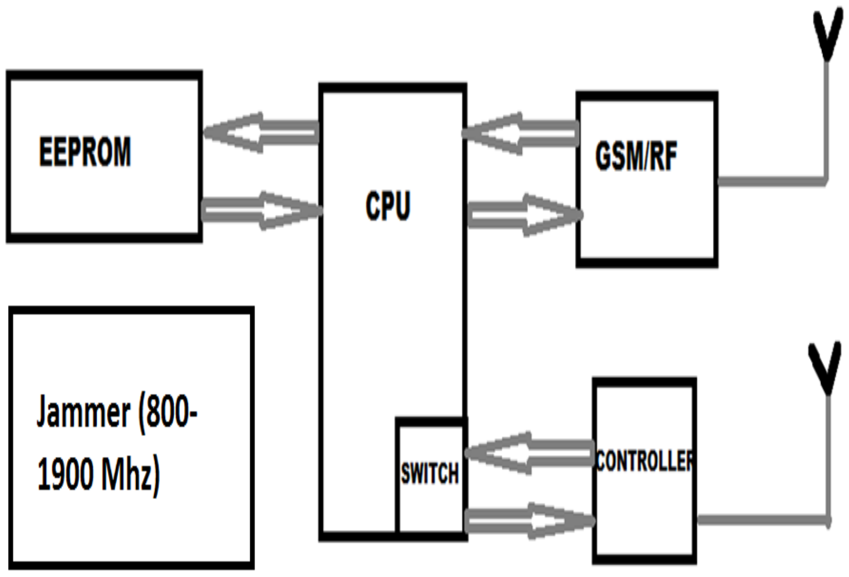


Figure 2.2: Circuit Diagram of Device1

2.2.2 NOVA INTELLIGENT DEVICE2

The nova intelligent device2 which is placed outside the restricted area just does the work of forwarding the call from the nova intelligent device1 to the respective base station in the case of an outgoing call and vice versa for an incoming call. In order to forward the call to the base station and to the device1 accordingly it has a separate connection with device1 and base station respectively.

Device1 is connected with Device2 via a separate communication channel with a different frequency, say 500MHZ. All the communication from the Device1 will always be first forwarded to Device2. Device2 has separate communication channel links with different Base stations of various Service providers in that area. For this to take place special authorization has to be obtained from the respective service providers. The channel can be with the usual frequency used by the Base Stations.

Thus, the nova intelligent device2 just consists of a GSM/RF ie, a transceiver and a receiver. The circuit diagram of our nova intelligent device2 is as follows:

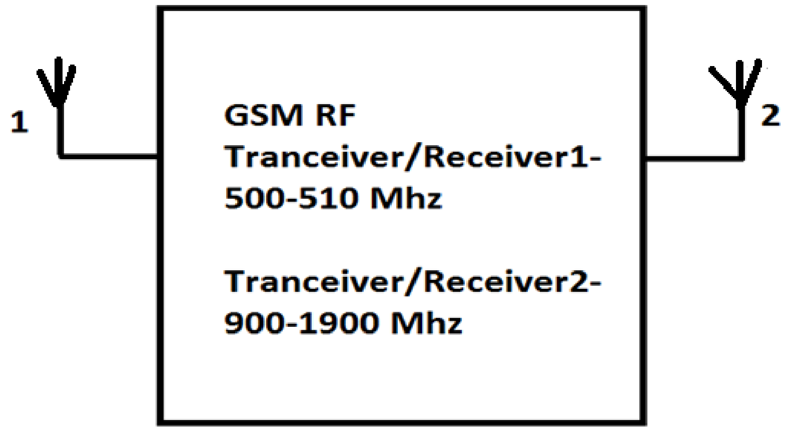


Figure 2.3: Circuit Diagram of Device2

Hence, for any incoming call/message, to a person in the restricted area, it will be via Device2 and then to Device1. If it's for an authorized number, then the communication is established. For an outgoing call in the restricted area, it will be via Device1. If it's from an authorized number, it will be forwarded to Device2 and from there to the Base station. Communication will thus be established.

2.2.3 SYSTEM IMPLEMENTATION

Our proposed system implementation involves the following steps.

Step 1: Jam the restricted area

A mobile phone jammer is an instrument used to prevent cellular phones from receiving signals from base stations. When used, the jammer effectively disables cellular phones. These devices can be used practically any location. Cell phone jammers block cell phone use by sending out radio waves along the same frequencies that cellular phones use. This causes enough interference with the communication between cell phones and Base Stations to render the phones unusable. On most retail phones, the network would simply appear out of range. Some of the images of a mobile phone jammer are as follows:

Figure 2.4 Jammers

These mobile phone jammers cost around \$160-\$180. These jammers are capable of jamming a distance of about 40-50 meters. The mobile jammers block all bands from 800 MHz to 1900 MHz within a particular range say, around 40-50 meters. So, depending upon the size of the restricted area the number of jammers to be used can be decided.

The main electronic components of a jammer are:
Voltage-controlled oscillator — Generates the radio signal that will interfere with the cell phone signal
Tuning circuit — Controls the frequency at which the jammer broadcasts its signal by sending a particular voltage to the oscillator
Noise generator — Produces random electronic output in a specified frequency range to jam the cell-phone network signal (part of the tuning circuit)
RF amplification (gain stage) — Boosts the power of the radio frequency output to high enough levels to jam a signal.

Therefore, the steps involved in jamming the restricted area are as follows:

1.1 Measure the restricted area.
1.2 Depending on the size of the restricted area decide the number of jammers to be used.
1.3 Hence jam the restricted area by placing the required number of jammers.

Step 2: Initiate the Nova device1

The next step is to design the nova intelligent device1. Our device1 is like a miniature base station which is specially designed for the restricted area and hence it consists of all the components similar to that of a base station. Base stations use radio signals to connect mobile devices to the network, enabling people to send and receive calls, texts, emails, pictures, web, TV and downloads. Without base stations, mobiles will not work.

Base stations are made up of three main elements:

- An antenna (or several antennas) to send and receive radio signals. These are typically between 0.5 and 2.5 meters long.
- A supporting structure such as a mast or building to hold the antenna(s) in the air.
- Equipment to power the base station and radio equipment, which is housed in protective cabinets.

In addition to these, our device also consists of some of the usual parts of a base station such as Power Amplifier, Combiner and Duplexer.

Apart from the base station components, our nova device also consists of an EEPROM to store the authorized numbers. **EEPROM** is an Electrically Erasable Programmable Read-Only Memory and is a type of non-volatile memory used in computers and other electronic devices to store data. EEPROM is user-modifiable read-only memory (ROM) that can be erased and reprogrammed (written to) repeatedly through the application of higher than normal electrical voltage generated externally or internally in the case of modern EEPROMs. Hence, our device1 makes use of this EEPROM to store the numbers of the authorized users. The main advantage of using an EEPROM is that it's convenient to change the list of authorized numbers at any instant.

Step 3: Attach Frequency Mixer to every authorized handset

The restricted area is jammed completely, so neither the authorized users nor the unauthorized users would be able to make or receive a call. In order to allow the authorized users to make or receive a call without any hindrance, a frequency mixer is attached to each and every authorized handset. The job of a frequency mixer is to shift signals from one frequency range to another, a process known as heterodyning, for convenience in transmission or further signal processing. The frequency mixer is found to be very economical as it costs just around $2.49.

The working of a frequency mixer can be depicted using the following figure:

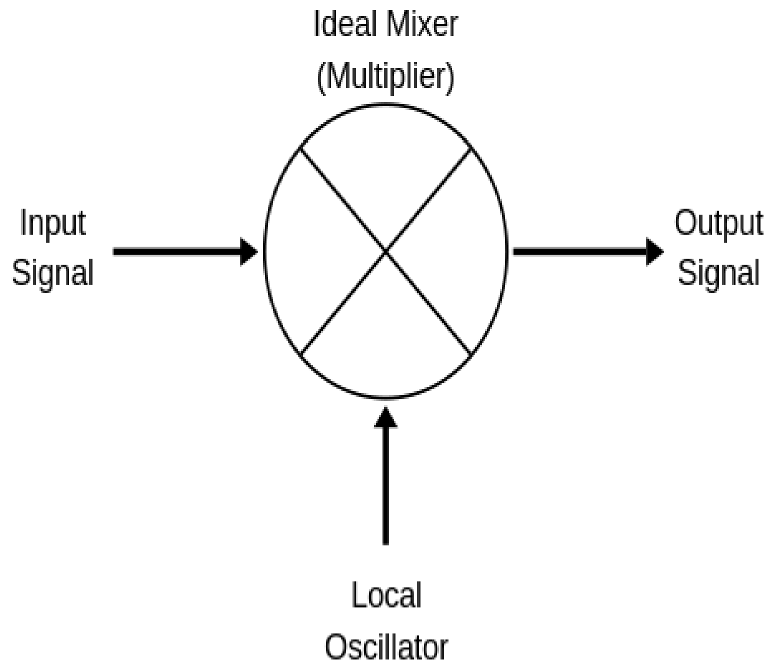


Figure 2.5: Frequency Mixer

Some of the salient features of a frequency mixer are as follows:

- Operating Temperature -40°C to 85°C
- RF Power 50mW
- IF Current 40mA
- Low conversion loss, 5.0 dB typ.
- Excellent L-R isolation, 55 dB typ.
- Excellent IP3, 15 dB m typ.
- Low profile package

When an authorized person tries to make a call he would not be able to do so, since the area is completely jammed. So in order to overcome this, the transceiver in the nova device1 sends signals with frequency 200-220MHZ. This localized frequency (200-220 MHz) has to be purchased from the **Federal Communication Commission (FCC).** The Federal Communications

Commission (FCC) is an independent agency of the United States government, created by Congressional statute, and with the majority of its commissioners appointed by the current President. The FCC works towards six goals in the areas of broadband, competition, the spectrum, the media, public safety and homeland security. The FCC was established by the Communications Act of 1934 and is charged with regulating interstate and international communications by radio, television, wire, satellite and cable.

Normally when the call option is pressed in a mobile, the mobile sends a signal in the range 900-1800MHZ frequency according to the SIM module embedded in the Mobile Phone. But, for the authorized numbers to use the 200-250MHZ frequency available in the jammed area, the frequency mixer used. The frequency mixer attached to the authorized handsets converts the frequency range of the signals generated by the mobile phone to our required range of 200-220MHz. Here, when a call is made from an authorized phone, the signals generated are in the frequency 200-250MHZ.

Step 4: Initiate the Nova Intelligent Device2

Device1 is connected with Device2 via a separate communication channel with a different frequency, say 500MHZ. This frequency range should also be purchased from the Federal Communications Commission (FCC).All the communication from the Device1 will always be first forwarded to Device2. Device2 has separate communication channel links with different Base stations of various Service providers in that area. This channel can be with the usual frequency used by the Base Stations.

Hence, for any incoming call/message, to a person in the restricted area, it will be via Device2 and then to Device1. If it’s for an authorized number, then the communication is established.

SCENARIO 1

Now, let's consider a scenario in which an authorized person who is inside the restricted area tries to make a call. The following actions take place when an authorized person attempts to make a call.

- When the call button is pressed, the frequency mixer which is attached to the authorized handset converts the frequency to a localized frequency say around 200-220MHz.
- This call is then forwarded to our nova intelligent device1, it then checks whether the number is authorized or not by comparing the number with the numbers stored in the EEPROM.
- When the number is found to be authorized, the nova intelligent device1 then forwards the call to the nova intelligent device2.
- Nova intelligent device2 then forwards the call to the respective base station through the separate communication link established between device2 and the base station.
- From there on the communication takes place in a usual manner, the reply then reaches the authorized person via device2 and device1This scenario can be depicted using block diagram as follows:

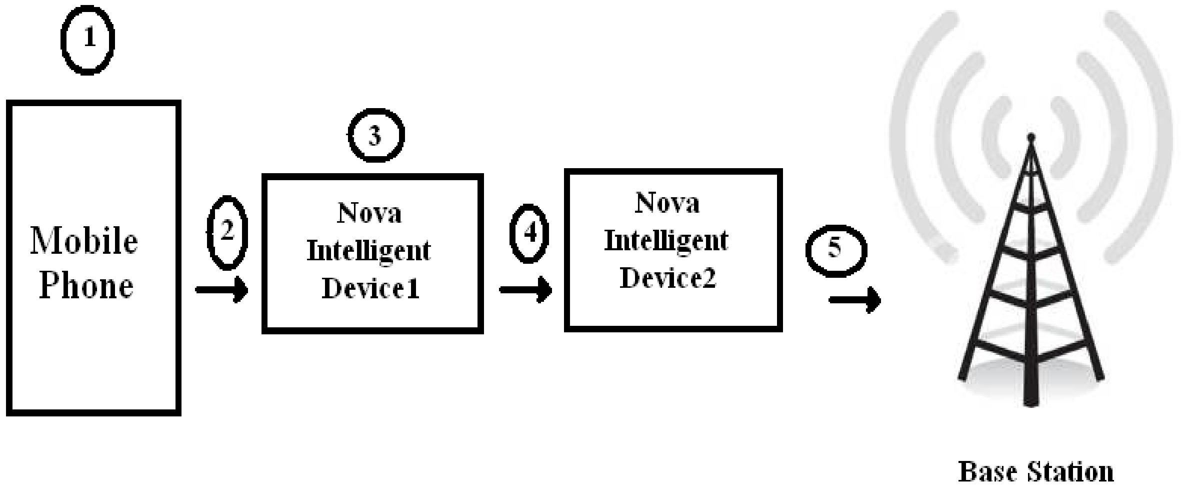


Figure 2.6: Outgoing call Concept

1-mobile phone makes a call

2-call is forwarded to device1

3-device1 checks whether the number is authorized

4-when the number is found to be authorized, call is forwarded to device2

5-device2 forwards the call to the respective base station.

2.3 Feasibility Study

A feasibility study assesses the operational, technical and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worth proceeding to the analysis phase. From the system analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

The feasibility study is a management-oriented activity. The objective of a feasibility study is to find out if an information system project can be done and to suggest possible alternative solutions.

A feasibility study should provide management with enough information to decide:

- Whether the project can be done
- Whether the final product will benefit its intended users and organization
- What are the alternatives among which a solution will be chosen
- Is there a preferred alternative

2.3.1 Economic Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with

costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.
Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs?
- The cost of doing full system study
- The cost of business employee time
- Estimated cost of hardware
- Estimated cost of software/software development
- Is the project possible, given the resource constraints?
- What are the savings that will result from the system?
- Cost of employees' time for study
- Cost of packaged software/software development
- Selection among alternative financing arrangements (rent/lease/purchase)

The concerned business must be able to see the value of the investment it is pondering before committing to an entire system study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, then the system is not economically feasible, and the project should not proceed any further. If the expected benefits equal or exceed costs, the system can be judged to be economically feasible. Economic analysis is used for evaluating the effectiveness of the proposed system.

The economical feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe. The required timeframe would need to be set by the organization.

In the case of our project, the various costs which the college would incur in implementing our project are as follows:

- Cost of a jammer:
  -a jammer costs around 10,000-15,000 depending upon the requirement.
- Cost of an EEPROM:
  -an EEPROM costs $2
- Authorization cost:
  -to implement our system we need to get authorization from all the service providers so we have to pay some amount to the service providers to get their approval.
- Cost of buying a localized frequency:
  -In order to make use of a localized frequency say 200-220 MHz, it has to be bought by the college from the FCC (Federal Communications Commission).

Thus, our proposed project is found to be economically feasible and beneficial when compared to the other existing systems.

2.3.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

The essential questions that help in testing the operational feasibility of a system include the following:

- Could there be increase in benefits?
- Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information?
- Does current mode of operation make maximum use of available resources, including people, time, and flow of forms?
- Does current mode of operation provide reliable services
- Are the services flexible and expandable?
- Are the current work practices and procedures adequate to support the new system?
- If the system is developed, will it be used?
- Manpower problems
- Labor objections
- Manager resistance
- Organizational conflicts and policies
- Social acceptability
- Government regulations
- Does management support the project?
- Will the proposed system really benefit the organization?
- Legal aspects
- Can or will end-users and management adapt to the change?

2.3.3 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements. The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfills the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility.

The essential questions that help in testing the operational feasibility of a system include the following:

- Is the project feasible within the limits of current technology?
- Does the technology exist at all?
- Is it available within given resource constraints?
- Is it a practical proposition?
- Manpower- programmers, testers & debuggers
- Software and hardware
- Are the current technical resources sufficient for the new system?
- Do we possess the necessary technical expertise, and is the schedule reasonable?
- Can the technology be easily applied to current problems?

CHAPTER 3
SYSTEM SPECIFICATION

3.1 SOFTWARE REQUIREMENTS

Operating System	:	Fedora 14, linux
Simulation Tool	:	NS-2
Documentation	:	Ms-Office

3.2HARDWARE REQUIREMENTS

CPU type	:	Intel Pentium 4
Clock speed	:	2.5 GHz
Ram size	:	512 MB
Hard disk capacity	:	80 GB
Monitor type	:	15 Inch color monitor
Keyboard type	:	Internet keyboard
CD -drive type	:	52xmax

CHAPTER 4
SOFTWARE DESCRIPTION

We are going to simulate our project idea using Network Simulator 2 Tool. We are creating many nodes representing mobile phones, base stations and our devices.

4.1 Network Simulator (ns2)

4.1.1 NS-2

Network Simulator (Version 2), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as Well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors. Network Simulator version 2 (NS-2) is a free and open source discrete event network simulator developed at UC Berkeley. NS is a discrete event simulator where the advance of time depends on the timing of events which are maintained by a scheduler. Ns is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. A network simulator is a piece of software or hardware that predicts the behavior of a network, without an actual network being present. NS-2 has a large and rich library of network and protocol objects. It covers a large part of applications (Ib, FTP, CBR,. . .), protocols (transport and routing protocols), network types (Satellite links, wired and wireless LAN), network elements (mobile nodes, wireless channel models, link and queue models,. . .) and traffic models (exponential, uniform, . . .). NS also allows to add and test new protocols and applications and/or to modify existing ones.

NS-2 is based on an object oriented simulator written in C++ and a OTCL interpreter (an object oriented extension of Tool Command Language TCL) These different objects are written in C++ code in order to achieve efficiency in the simulation and faster execution times. (e.g. Implementation of IEEE 802.11 is found in . . . ns-**/ns-**/mac/802_11.cc,h) The OTCL script, which is provided by the user at the simulation time, is used to define and to configure the network topology and network elements (node type, the protocols and applications to be used), and to schedule the events. The OTCL scripts are used also to tell NS to create a visualization trace as Well as an ascii file trace corresponding to the events generated in the network. To analyses the trace files, other independent tools will be needed to filter, compute and display the results (e.g. Awk, Matlab, gnuplot, etc.). Two other independent and optional tools are provided with NS packages: Network animator (Nam) and xgraph. OTCL scripts can be written in any text editor like kateor emacs.

4.1.2 BASIC ARCHITECTURE

Figure shows the basic architecture of NS2. NS2 provides users with an executable command n which takes on input argument, the name of a TCL simulation scripting file. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTCL). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTCL sets up simulation by assembling and configuring the objects as Well as scheduling discrete events (i.e., a frontend). The C++ and the OTCL are linked together using +-. Mapped to a C++ object, variables in the OTCL domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g., _o10) in the OTCL domain, and does not contain any functionality. Instead, the functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTCL domain, a handle acts as a frontend which interacts with users and other OTCL objects. It may define its own procedures and variables to facilitate the interaction. Note that the member procedures and variables in the OTCL domain are called instance procedures (instprocs) and instance variables (instvars), respectively [15].

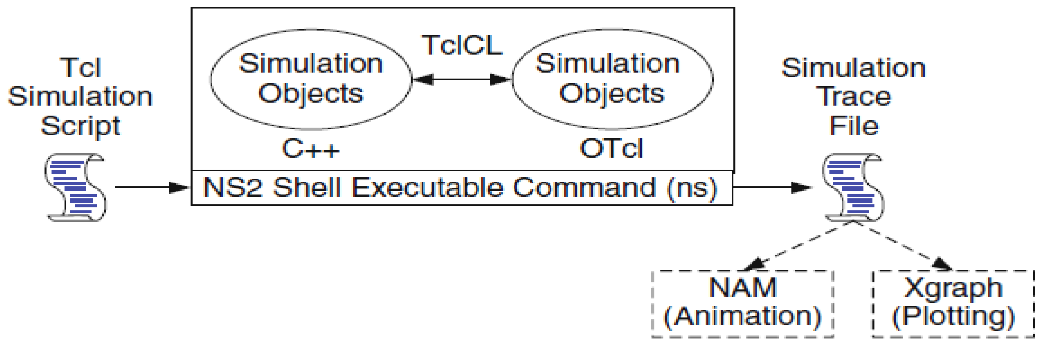


Figure 4.1: Basic Architecture of NS

NS2 provides a large number of built-in C++ objects. It is advisable to use these C++ objects to set up a simulation using a TCL simulation script. However, advance users may find these objects insufficient. They need to develop their own C++ objects, and use anOTCL configuration interface to put together these objects. After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network AniMator) and XGraph are used. To analyze a particular behavior of the network, users can extract a relevant subset of text-based data and transform it to a more conceivable presentation.

4.1.3 NS2 SIMULATION STEPS

The followings show the three key step guideline in defining a simulation scenario in a NS2:

Step 1: Simulation Design

The first step in simulating a network is to design the simulation. In this step, the users should determine the simulation purposes, network configuration and assumptions, the performance measures, and the type of expected results.

Step 2: Configuring and Running Simulation

This step implements the design in the first step. It consists of two phases:

• **Network configuration phase**: In this phase network components (e.g., node, TCP and UDP) are created and configured according to the simulation design. Also, the events such as data transfer are scheduled to start at a certain time.

• **Simulation Phase**: This phase starts the simulation which was configured in the Network Configuration Phase. It maintains the simulation clock and executes events chronologically.

Step 3: Post Simulation Processing

The main tasks in this step include verifying the integrity of the program and evaluating the performance of the simulated network. While the first task is referred to as debugging, the second one is achieved by properly collecting and compiling simulation results.

4.1.3.1SIMULATION DESIGN

Figure shows the configuration of a network under consideration. The network consists of five nodes n0 to n4. In this scenario, node n0 sends constant bit rate (CBR) traffic to node n3, and node n1 transfers data to node n4 using a file transfer protocol (FTP).

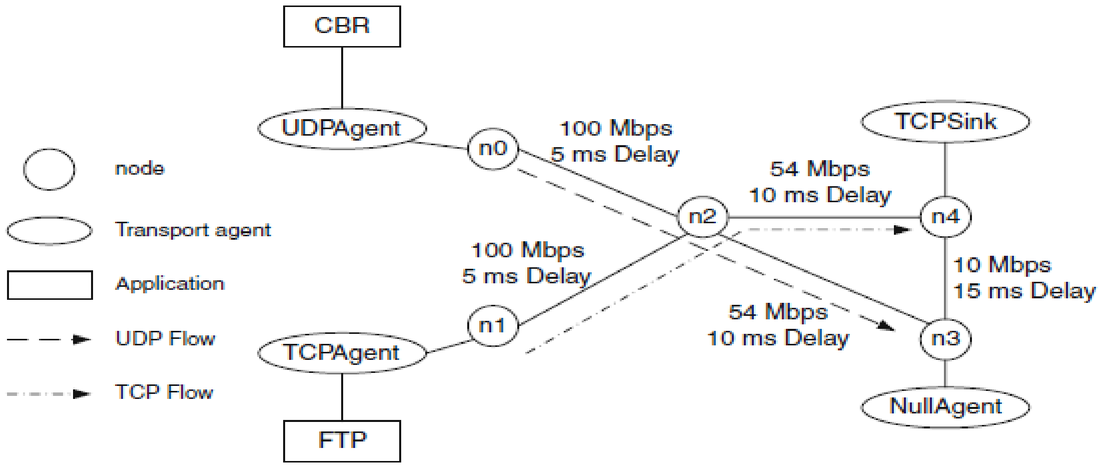


Figure 4.2: Sample Network Topology

4.1.3.2NETWORK ANIMATION (NAM) TRACE

NAM trace is records simulation detail in a text file, and uses the text file the play back the simulation using animation. NAM trace is activated by the command “$ns namtrace-all $file”, where ns is the Simulator handle and file is a handle associated with the file (e.g., out.nam in the above example) which stores the NAM trace information. After obtaining a NAM trace file, the animation can be initiated directly at the command prompt through the following command:

```
>>namfilename.nam
```

Many visualization features are available in NAM. These features are for example animating colored packet flows, dragging and dropping nodes (positioning), labeling nodes at a specified instant, shaping the nodes, coloring a specific link, and monitoring a queue.

4.1.4 LINKAGE BETWEENOTCL AND C++ IN NS2

NS2 is an object oriented simulator written in OTCL and C++ languages. While OTCL acts as the frontend (i.e., user interface), C++ acts as the backend running the actual simulation. As can be seen from Fig, class hierarchies of both languages can be either standalone or linked together using an OTCL/C++ interface called TCL . There are two types of classes in each domain. The first type includes classes which are linked between the C++ and OTCL domains. In the literature, these OTCL and C++ class hierarchies are referred to as the interpreted hierarchy and the compiled hierarchy, respectively. The second type includes OTCL and C++ classes which are not linked together [15]. These classes are neither a part of the interpreted hierarchy nor a part of compiled hierarchy. This chapter discusses how OTCL and C++ languages constitute NS2.

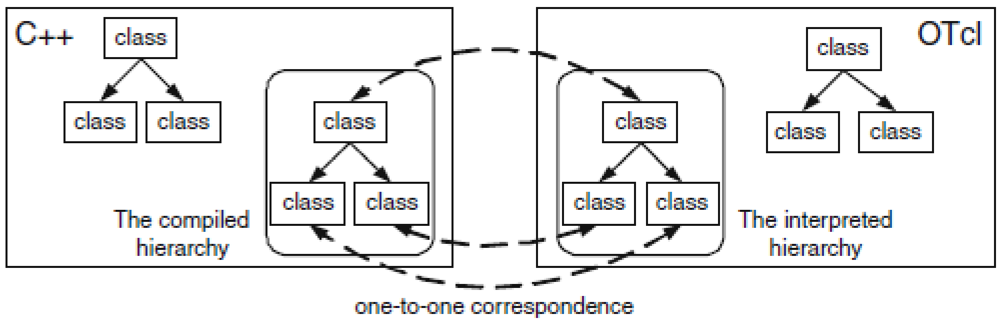


Figure 4.3: Two language structure of NS2

Class hierarchies in both the languages may be standalone or linked together. OTCL and C++ class hierarchies which are linked together are called the interpreted hierarchy and the compiled hierarchy, respectively.

4.1.5 THE TWO-LANGUAGE CONCEPT IN NS2

Why two languages? NS2 uses OTCL to create and configure a network, and uses C++ to run simulation. All C++ codes need to be compiled and linked to create an executable file. Since the body of NS2 is fairly large, the compilation time is not negligible. A typical Pentium 4 computer requires few seconds (long enough to annoy most programmers) to compile and link the codes with a small change such as including "int i=0;" into the codes. OTCL, on the other hand, is an interpreter, not a compiler. Any change in a OTCL file does not need compilation. Nevertheless, since OTCL does not convert all the codes into machine language, each line needs more execution time. In summary, C++ is fast to run but slow to change. It is suitable for running a large simulation. OTCL, on the other hand, is slow to run but fast to change. It is therefore suitable to run a small simulation over several repetitions (each may have different parameters). NS2 is constructed by combining the advantages of these two languages.

NS2 provides the following guidelines to choose a coding language:

• Use OTCL

– For configuration, setup, or one time simulation, or

– To run simulation with existing NS2 modules.

This option is preferable for most beginners, since it does not involve complicated internal mechanism of NS2. Unfortunately, existing NS2 modules are fairly limited. This option is perhaps not sufficient for most researchers.

• Use C++

– When you are dealing with a packet, or when you need to modify existing NS2 modules. This option perhaps discourages most of the beginners from using NS2.

4.1.6 IMPLEMENTATION OF DISCRETE-EVENT SIMULATION IN NS2

NS2 is a discrete-event simulator, where actions are associated with events rather than time. An event in a discrete-event simulator consists of execution time, a set of actions, and a reference to the next event. These events connect to each other and form a chain of events on the simulation timeline. Unlike a time-driven simulator, in an event-driven simulator, time between a pair of events does not need to be constant. When the simulation starts, events in the chain are executed from left to right (i.e., chronologically).

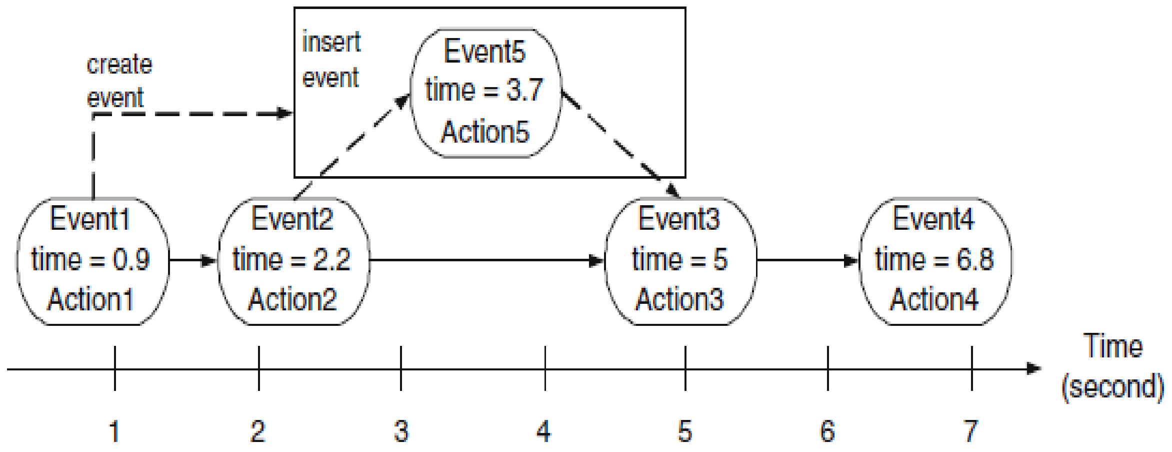

create event
insert event
Event5 time = 3.7 Action5
Event1 time = 0.9 Action1
Event2 time = 2.2 Action2
Event3 time = 5 Action3
Event4 time = 6.8 Action4
Time (second)
1
2
3
4
5
6
7

Figure 4.4: Sample chain of events in discrete-event simulation

4.1.6.1 NS2 SIMULATION CONCEPT

NS2 simulation consists of two major phases.

Phase I: Network Configuration Phase

In this phase, NS2 constructs a network and sets up an initial chain of events. The initial chain of events consists of events which are scheduled to occur at certain times (e.g., start FTP (File Transfer Protocol) traffic at 1 second.). These events are called at-events. This phase corresponds to every line in a TCL simulation script before executing instprocrun{} of the Simulator object.

Phase II: Simulation Phase

This part corresponds to a single line, which invokes instproc Simulator::run {}. Ironically, this single line contributes to most (e.g., 99%) of the simulation. In this part, NS2 moves along the chain of events and executes each event chronologically. Here, the instproc Simulator::run{} starts the simulation by dispatching the first event in the chain of events. In NS2, "dispatching an event" or "firing an event" means "taking actions corresponding to that event". An action is, for example, starting FTP traffic or creating another event and inserting the created event into the chain of events. In Fig., at 0.9 s, Event1 creates Event5 which will be dispatched at 3.7 s, and inserts Event5 after Event2. After dispatching an event, NS2 moves down the chain and dispatches the next event. This process repeats until the last event corresponding to instprochalt{} of OTCL class Simulator is dispatched, signifying the end of simulation.

4.1.7 OVERVIEW OF NS2 COMPONENTS

Based on the functionality, NS2 modules (or objects) can be classified into four following types:

• **Network objects**are responsible for sending, receiving, creating, and destroying packet-related objects. Since these objects are those derived from class NsObject, they will be referred to hereafter as NsObjects.

• **Packet-related objects**are various types of packets which are passed around a network.

• **Simulation-related objects**control simulation timing, and supervise the entire simulation. As

Examples of simulation-related objects are events, handlers, the Scheduler, and the Simulator.

• **Helper objects**do not explicitly participate in packet forwarding. However, they implicitly help to complete the simulation. For example, a routing module calculates routes from a source to a destination, while network address identifies each of the network objects.

4.1.8 PACKET FORWARDING MECHANISM OF NSOBJECTS

An NsObject forwards packets in two following ways:

• **Immediate packet forwarding**: To forward a packet to a downstream object, an upstream object needs to obtain a reference (e.g., a pointer) to the downstream object and invokes function recv(p,h) of the downstream object through the obtained reference. For example, a Connector has a private pointer target_ to its downstream object. Therefore, it forwards a packet to its downstream object by invoking
target_->recv(p,h).

• **Delayed packet forwarding**: To delay packet forwarding, a Packet object is cast to be an Event object, associated with a packet receiving NsObject, and placed on the simulation timeline at a given simulation time. At the firing time, function handle (e) of the NsObjectwill be invoked, and the packet will be received through function recv(p,h).

4.1.9 AN OVERVIEW OF NODES IN NS2

A Node plays two important roles in NS2. As a router, it forwards packets to the connecting link based on a routing table. As a host, it delivers packets to the transport layer agent attached to the port specified in the packet header. NS2 configures the connection to its downstream NsObjects only [16]. A Node does not need to have a connection to its upstream NsObject (e.g., a sending transport agent or an upstream link). Instead, its upstream NsObjectwill create a connection to the Node.

4.1.9.1 ARCHITECTURE OF A NODE

In the OTCL domain, a Node is defined in a C++ class Node which is bound to an OTCL class with the same name. A Node is a composite object whose architecture is shown in Fig. It provides a single point of packet entrance, entry (which is a Connector object). After entering the Node entry, the packet enters an address classifier (an instvar classifier). If the Node is not the final destination, the address classifier will forward the packet to the link specified in the routing table. Otherwise, it will forward the packet to the demultiplexer or port classifier (an instvardmux), which forwards the packet to the agent attached to the port specified in the packet header.

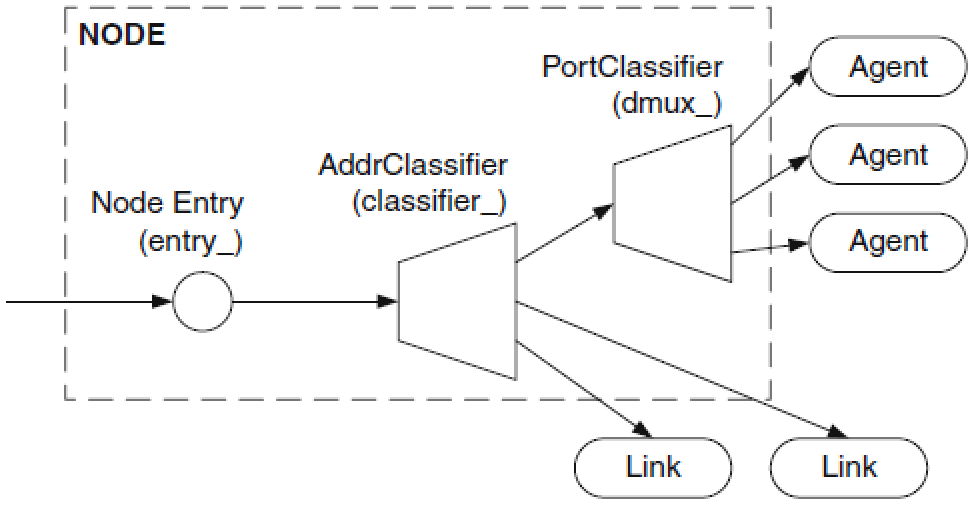


Figure 4.5: Node Architecture

4.2 Features

Network simulators serve a variety of needs. Compared to the cost and time involved in setting up an entiretest bed containing multiple networked computers, routers and data links, network simulators are relatively fast and inexpensive. They allow engineers, researchers to test scenarios that might be particularly difficult or expensive to emulate using real hardware - for instance, simulating a scenario with several nodes or experimenting with a new protocol in the network. Network simulators are particularly useful in allowing researchers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment.

Network simulators, as the name suggests are used by researchers, developers and engineers to design various kinds of networks, simulate and then analyze the effect of various parameters on the network performance. A typical network simulator encompasses a wide range of networking technologies and can help the users to build complex networks from basic building blocks such as a variety of nodes and links. With the help of simulators, one can design hierarchical networks using various types of nodes like computers, hubs, bridges, routers, switches, links, mobile units etc.

Various types of Wide Area Network (WAN) technologies like TCP, ATM, IP etc and Local Area Network (LAN) technologies like Ethernet, token rings etc., can all be simulated with a typical simulator and the user can test, analyze various standard results apart from devising some novel protocol or strategy for routing etc.

There are a wide variety of network simulators, ranging from the very simple to the very complex. Minimally, a network simulator must enable a user to represent a network topology, specifying the nodes on the network, the links between those nodes and the traffic between the nodes. More complicated systems may allow the user to specify everything about the protocols used to handle network traffic. Graphical applications allow users to easily visualize the workings of their simulated environment. Text-based applications may provide a less intuitive interface, but may permit more advanced forms of customization. Others, such as GTNets, are programming-oriented, providing a programming framework that the user then customizes to create an application that simulates the networking environment to be tested.

4.2.1 SUMMARY

This chapter has demonstrated how new modules are created, configured, and incorporated into NS2. In most of the cases, I need to develop NS2 codes in both C++ and OTCL domains. In the C++ domain, the main task is to define the internal mechanisms of the new NS2 components. The main task in the OTCL domain, on the other hand, is to integrate the developed NS2 components into the existing NS2 modules and to instantiate and configure the newly developed modules from a TCL simulation script.

Chapter 5
PROJECT DESCRIPTION

5.1 Problem Definition

The main objective of our project is to design a “Nova Intelligent Device”, which classifies the authorized and unauthorized numbers in a particular area and allows communication establishment to only the authorized numbers and blocks all unauthorized users. The key feature in this device is that, we are using a jammer, which will block all the signals from or to any base station to a mobile phone. Here, we make use of two Nova intelligent devices, where one is inside the jammed area, named as device1 and another is outside the jammed area, named as device2. Another point to be noted here is that, there is already a separate channel allocated between our device2 and the base stations of different service providers in that area, with prior authorization from the respective service providers. For any incoming call to a person in the restricted area, the call would be first directed to our Device2 via the separate channel assigned. Then the Device2 would forward the call to Device1 which is inside the jammed area. For any outgoing call, the call would be first directed to our device1 and then to the device2 and then to the respective base station if in case the call is made by an authorized person.

5.2 Overview of Project

The overview of our project is to design a “Nova Intelligent Device”, which classifies the authorized and unauthorized numbers in a particular area and allows communication establishment to only the authorized numbers and blocks all unauthorized users. The key feature in this device is that, we are using a jammer, which will block all the signals from or to any base station to a mobile phone. Here, we make use of two Nova intelligent devices, where one is inside the jammed area, named as device1 and another is outside the jammed area, named as device2.There is already a separate channel allocated between our device2 and the base stations of different service providers in that area, with prior authorization from the respective service providers. For any incoming call to a person in the restricted area, the call would be first directed to our Device2 via the separate channel assigned.The Device2 would forward the call to Device1 which is inside the jammed area. For any outgoing call, the call would be first directed to our device1 and then to the device2 and then to the respective base station if in case the call is made by an authorized person.

5.3 Module Description

We are going to simulate this project using Network Simulator II.

MODULE 1: ***Creation of nodes***

In this first module, we created a total of 14 nodes, where, 10 nodes represent mobile phone, 2 nodes represent base and 2 nodes represent device1 and device2.

MODULE 2: ***Communication between the mobile nodes and our device for outgoing call***

In our second module, we establish communication between our mobile phone nodes and the devices. The device forwards the call to the base station, if it's authorized.

If it's an authorized mobile node trying to establish communication, then the color is changed to green and communication takes place.

If it's an unauthorized mobile node trying to establish communication, then the color is changed to Red and communication is denied.

MODULE 3: ***Communication between the mobile nodes and our device for incoming call***

If it's for an authorized mobile, then mobile phone node color is changed to green and communication takes place. If it's for an unauthorized mobile, then color is changed to Red and communication is denied.

.

CHAPTER VI
SYSTEM IMPLEMENTATION

6.1 Source Code

```
set val(chan)          Channel/WirelessChannel    ;# channel type
set val(prop)          Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)         Phy/WirelessPhy            ;# network interface type
set val(mac)           Mac/802_11                 ;# MAC type
set val(ifq)           Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)            LL                         ;# link layer type
set val(ant)           Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)        50                         ;# max packet in ifq
set val(nn)            14                         ;# number of mobilenodes
set val(rp)            DumbAgent                  ;# routing protocol
set val(x)             1000
set val(y)             1000
set val(stop)          70

#-------Event scheduler object creation--------#

set ns [new Simulator]

# Creating trace file and nam file

set tracefd [open try.tr w]
set namtrace [open try.nam w]

$ns trace-all $tracefd
$ns namtrace-all-wireless $namtrace $val(x) $val(y)
```

```
# set up topography object

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]

# configure the nodes

        $ns node-config -adhocRouting $val(rp) \
                   -llType $val(ll) \
                   -macType $val(mac) \
                   -ifqType $val(ifq) \
                   -ifqLen $val(ifqlen) \
                   -antType $val(ant) \
                   -propType $val(prop) \
                   -phyType $val(netif) \
                   -channelType $val(chan) \
                   -topoInstance $topo \
                   -agentTrace ON \
                   -routerTrace ON \
                   -macTrace OFF \
                   -movementTrace ON



# Creating node objects...
for {set i 0} {$i < $val(nn) } { incr i } {
            set node_($i) [$ns node]
      }
```

```
# for {set i 0} {$i < $val(nn)  } {incr i } {
#      $node_($i) color blue
#     $ns at 0.0 "$node_($i) color blue" }


# node color and label

$node_(0) color black
$ns at 0.0 "$node_(0) color black"
#$ns at 0.0 "$node_(0) label M"

$node_(0) color orange
$ns at 28.0 "$node_(0) color orange"

$node_(1) color black
$ns at 0.0 "$node_(1) color black"

$node_(2) color black
$ns at 0.0 "$node_(2) color black"
#$ns at 0.0 "$node_(0) label M"

$node_(3) color black
$ns at 0.0 "$node_(3) color black"

$node_(3) color green
$ns at 2.0 "$node_(3) color green"
```

```
$node_(4) color black
$ns at 0.0 "$node_(4) color black"

$node_(4) color red
$ns at 14.0 "$node_(4) color red"


$node_(5) color black
$ns at 0.0 "$node_(5) color black"
$node_(5) color red

$ns at 17.0 "$node_(5) color red"
#$ns at 0.0 "$node_(0) label M"

$node_(6) color black
$ns at 0.0 "$node_(6) color black"

$node_(6) color green
$ns at 20.0 "$node_(6) color green"


$node_(7) color black
$ns at 0.0 "$node_(7) color black"

$node_(7) color green
$ns at 32.0 "$node_(7) color green"

$node_(8) color blue
$ns at 0.0 "$node_(8) color blue"
$ns at 0.0 "$node_(8) label D1"
```

```
$node_(9) color blue
$ns at 0.0 "$node_(9) color blue"
$ns at 0.0 "$node_(9) label D2"

$node_(10) color purple
$ns at 0.0 "$node_(10) color purple"
$ns at 0.0 "$node_(10) label Base_Station1"

$node_(11) color purple
$ns at 0.0 "$node_(11) color purple"
$ns at 0.0 "$node_(11) label Base_Station2"



$node_(12) color orange
$ns at 0.0 "$node_(12) color orange"
#$ns at 0.0 "$node_(0) label M"



$node_(13) color orange
$ns at 0.0 "$node_(13) color orange"
#$ns at 0.0 "$node_(0) label M"

# Provide initial location of mobile nodes

$node_(0) set X_ 234.26
$node_(0) set Y_ 151.394
$node_(0) set Z_ 0.0

$node_(1) set X_ 350.0
$node_(1) set Y_ 100.0
$node_(1) set Z_ 0.0
```

```
$node_(2) set X_ 570.972
$node_(2) set Y_ 166.078
$node_(2) set Z_ 0.0

$node_(3) set X_ 625.527
$node_(3) set Y_ 344.052
$node_(3) set Z_ 0.0

$node_(4) set X_ 340.924
$node_(4) set Y_ 348.266
$node_(4) set Z_ 0.0


$node_(5) set X_ -7.30918
$node_(5) set Y_ 477.974
$node_(5) set Z_ 0.0

$node_(6) set X_ 200.0
$node_(6) set Y_ 500.0
$node_(6) set Z_ 0.0

$node_(7) set X_ 825.801
$node_(7) set Y_ 410.13
$node_(7) set Z_ 0.0

$node_(8) set X_ 462.577
$node_(8) set Y_ 490.891
$node_(8) set Z_ 0.0

$node_(9) set X_ 472.366
```

```
$node_(9) set Y_ 690.211
$node_(9) set Z_ 0.0

$node_(10) set X_ 634.309
$node_(10) set Y_ 839.295
$node_(10) set Z_ 0.0

$node_(11) set X_ 295.571
$node_(11) set Y_ 838.968
$node_(11) set Z_ 0.0

$node_(12) set X_ 804.214
$node_(12) set Y_ 995.104
$node_(12) set Z_ 0.0

$node_(13) set X_ 120.47
$node_(13) set Y_ 937.632
$node_(13) set Z_ 0.0

# Define node initial position in nam

for {set i 0} {$i < 8} { incr i } {
$ns initial_node_pos $node_($i) 60
}

for {set i 8} {$i < 10} { incr i } {
$ns initial_node_pos $node_($i) 100
}

$ns initial_node_pos $node_(10) 150
$ns initial_node_pos $node_(11) 150
```

```
$ns initial_node_pos $node_(12) 60
$ns initial_node_pos $node_(13) 60

#destination of nodes

$ns at 2.0 "$node_(3) setdest 808.211 195.469 30.0"
$ns at 5.0 "$node_(4) setdest 599.118 348.266 35.0"
$ns at 5.0 "$node_(5) setdest 389.725 361.056 35.0"
$ns at 10.0 "$node_(6) setdest 282.817 470.771 30.0"
$ns at 13.0 "$node_(7) setdest 672.346 485.641 25.0"
$ns at 17.0 "$node_(0) setdest 107.599 594.708 30.0"
$ns at 18.0 "$node_(4) setdest 331.196 175.325 25.0"
$ns at 20.0 "$node_(1) setdest 130.776 299.735 30.0"

# communication establihment



set k 3
set j 0

while {$k < 8}  {

set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_($k) $tcp
$ns attach-agent $node_(8) $sink
$ns connect $tcp $sink
$tcp set packetsize_ 10000
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

```
$ns at $j "$ftp start"
incr j
incr j
$ns at $j "$ftp stop"
incr j

# using a file to retrieve the the names of the nodes that are authorized

set fp [ open "/home/raji/ns-allinone-2.35/ns-2.35/tcl/ex/some.txt" "r" ]
fconfigure $fp -buffering line
gets $fp data

# making use of a while and if loop to check conditions and to retrieve the nodes from the
file
while { $data != "" } {

if {$data == "node_($k)"} {

set tcp1 [new Agent/TCP/Newreno]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(8) $tcp1
$ns attach-agent $node_(9) $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetsize_ 10000
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at $j "$ftp1 start"
incr j
incr j
$ns at $j "$ftp1 stop"
```

```
incr j

set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(9) $tcp
$ns attach-agent $node_(10) $sink
$ns connect $tcp $sink
$tcp set packetsize_ 10000
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at $j "$ftp start"
incr j
incr j
$ns at $j "$ftp stop"
incr j

set tcp1 [new Agent/TCP/Newreno]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(10) $tcp1
$ns attach-agent $node_(12) $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetsize_ 10000
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at $j "$ftp1 start"
incr j
incr j
$ns at $j "$ftp1 stop"
incr j
```

```
}

gets $fp data
}
incr k
close $fp
}

# incoming call

set tcp1 [new Agent/TCP/Newreno]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(13) $tcp1
$ns attach-agent $node_(11) $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetsize_ 10000
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 42.0 "$ftp1 start"
$ns at 44.0 "$ftp1 stop"

set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(11) $tcp
$ns attach-agent $node_(9) $sink
$ns connect $tcp $sink
$tcp set packetsize_ 10000
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

```
$ns at 44.1 "$ftp start"
$ns at 46.0 "$ftp stop"

set tcp1 [new Agent/TCP/Newreno]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(9) $tcp1
$ns attach-agent $node_(8) $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetsize_ 10000
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 46.1 "$ftp1 start"
$ns at 48.0 "$ftp1 stop"

set tcp1 [new Agent/TCP/Newreno]
$tcp1 set class_ 2
set sink1 [new Agent/TCPSink]
$ns attach-agent $node_(8) $tcp1
$ns attach-agent $node_(7) $sink1
$ns connect $tcp1 $sink1
$tcp1 set packetsize_ 10000
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 48.1 "$ftp1 start"
$ns at 50.0 "$ftp1 stop"


# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn) } { incr i } {
   $ns at $val(stop) "$node_($i) reset";
}

# Ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 70.01 "puts \"end simulation\"; $ns halt"
#stop procedure:
proc stop {} {
   global ns tracefd namtrace
   $ns flush-trace
   close $tracefd
   close $namtrace
exec nam try.nam &
}
$ns run
```

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Testing is the process of evaluating a system or its components to find whether it satisfies the specified requirements or not. It evaluates the features of the system and assesses the quality of the software product. It is a verification and validation process. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or Weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished. An early start to testing reduces the cost, time to rework and error free software that is delivered to the client. Testing is done in different forms at every phase of SDLC like during Requirement gathering phase, the analysis and verifications of requirements are also considered testing. Reviewing the design in the design phase with intent to improve the design is also considered as testing. Unlike when to start testing it is difficult to determine when to stop testing, as testing is a never ending process and no one can say that any software is 100% tested.

7.2 Unit testing

Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. In procedural programming a unit may be an individual program, function, procedure, etc. Unit Testing is performed by using the White Box Testing method. Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated.

It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

In our project we have three modules. Hence we have three units of code to be tested. Another to be noted is we have the coding done using the Network Simulator (ns2). As mentioned we have used the method of white box testing for testing the units of code.

Unit 1: The unit one is the code for the creation of nodes. In case of ns2 we have to declare the number of nodes we want to create prior to the code for creating it. Hence when this part was tested the observation was, we were able to create only the declared number of nodes and not more than that.

Unit 2: In this case the piece of code to be tested was the code for the communication between the nodes and the device (outgoing call). We have two possible cases of results in this. They are as follows:

Case 1 – In case of our coding part we have created a file with a list of authorized numbers. Hence if any authorized number is trying to make a call it will be allowed.

Case 2 – On the other hand if any unauthorized number is making a call it will be denied.

Hence the flow of control is tested for both the paths.

Unit 3 : In this case the piece of code to be tested was the code for the communication between the nodes and the device (incoming call) . We have two possible cases of results in this. They are as follows:

Case 1 – In case of our coding part we have created a file with a list of authorized numbers. Hence if the call is for an authorized number it will be allowed.

Case 2 – On the other hand if the call is for an unauthorized number it will be denied.

Hence the flow of control is tested for both the paths. And the unit testing is done.

7.3 Integration Testing

The purpose of this level of testing is to expose faults in the interaction between integrated units. Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

In this case, we have used the method of Black Box testing. In case of black box testing we have different techniques that can be adopted. We have chosen the Equivalence partitioning method. Equivalence Partitioning is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.

In our project the details of the test data are as follows:

Valid Input	**:** Nodes inside the restricted area which are allowed to communicate with our nova device1. (nodes declared)
Invalid Input	**:** Nodes outside the restricted area. (nodes undeclared)
Output	**:** Only the nodes that were inside the restricted area was able to communicate. The others were not able to established. Another point is only the authorized nodes were able to make/ receive call and the unauthorized nodes were not able to do so.

7.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. Normally, independent Testers perform System Testing.

In case of our project when the system testing was done using the method of black box testing the expected outputs were obtained. This shows the integration of the modules was proper and the software system was ready for the acceptance.

7.5Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered. Hence the different types of testing is done.

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENTS

In today's scenario mobile phones are very much essential. If we consider the case of colleges, the students should not use the mobile phones inside the college. We can use a metal detector to detect the mobile phones and stop the students from bringing it. But they need it when they go out of the college. Hence our device will prove to be useful in this case because it will allow them to bring the mobile phones and stop them from using it in restricted areas. Hence our system can also be extended to various other restricted areas like factories, hospitals and other places where such need exists. Thereby we conclude that our system will prove to be very efficient to prevent the illegal usage of mobile phones.

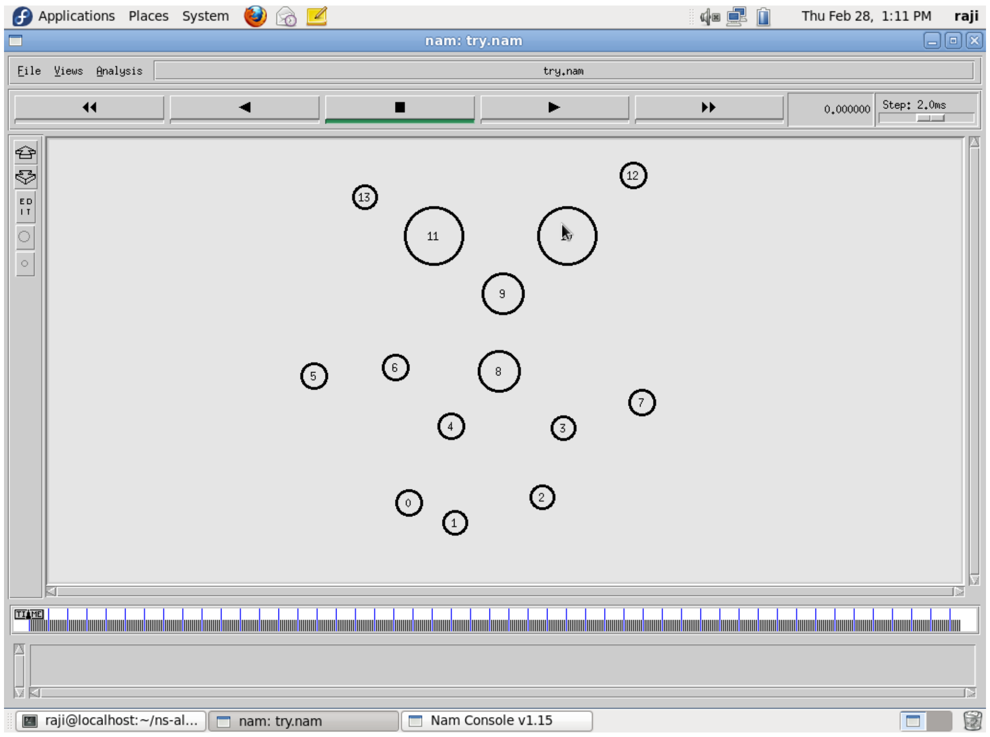
8.1 Future Enhancements

The idea that we have presented in our project is entirely new. Hence efforts are being taken in order to implement the project in real time. But, to implement the project in real time we need the help of the different service providers. We have to get prior authorization from the different service providers to complete the implementation successfully.

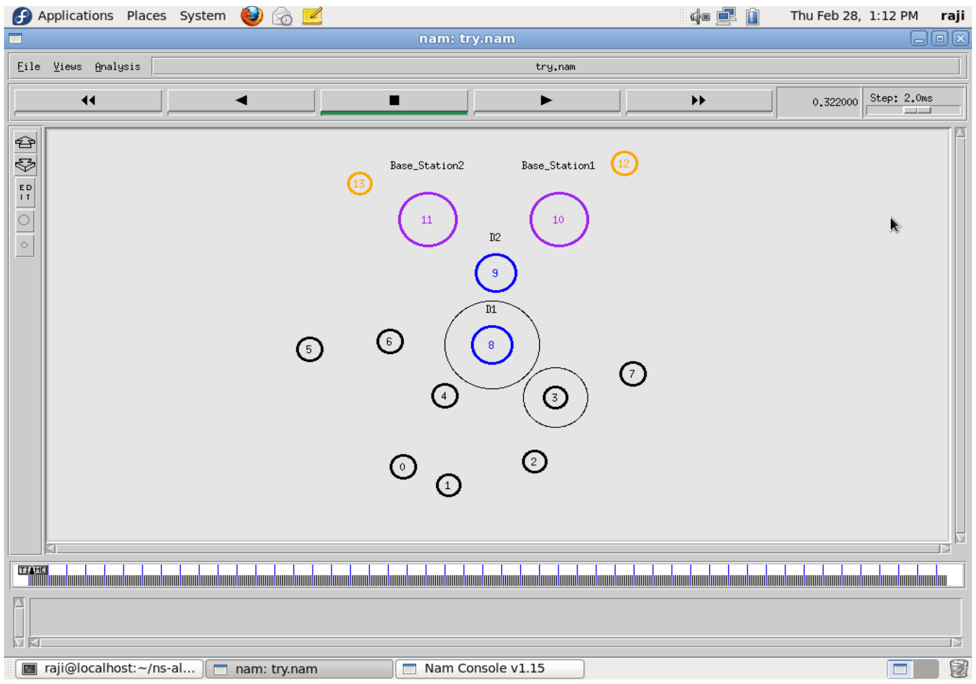
In our idea we need jammers to block the signals from the base station from reaching the mobile phones directly. The number of jammer depends on the area that is to be covered. Hence work can be made on the working of jammer so that the number of jammers needed can be reduced and the project will be much more cost efficient.

CHAPTER 9
APPENDIX

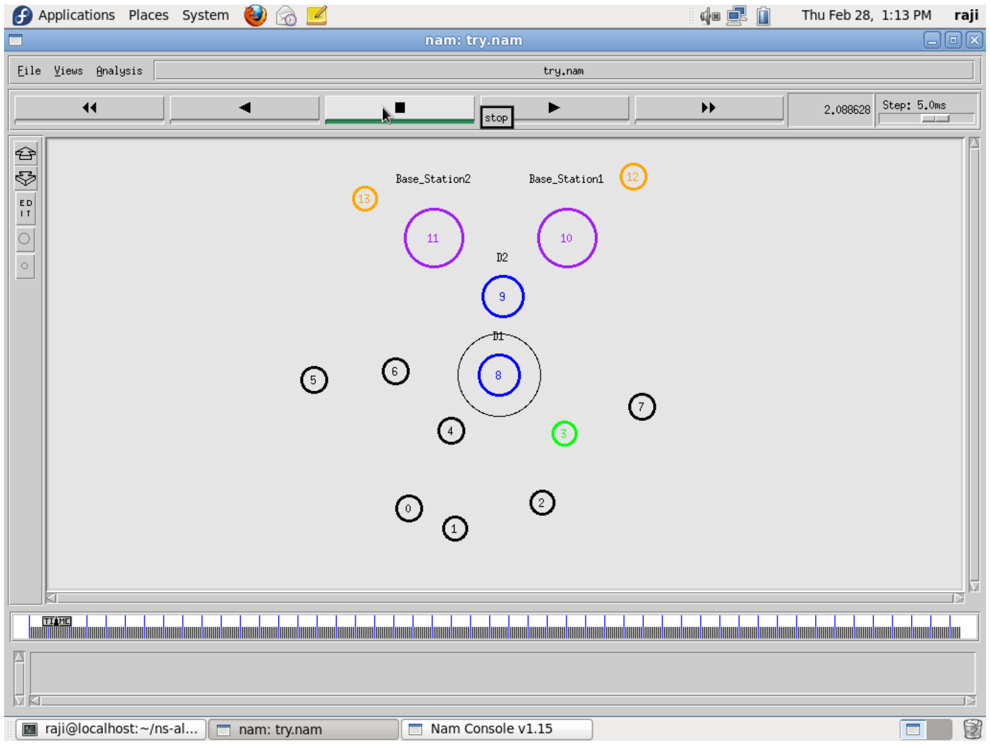
9.1 Screenshots



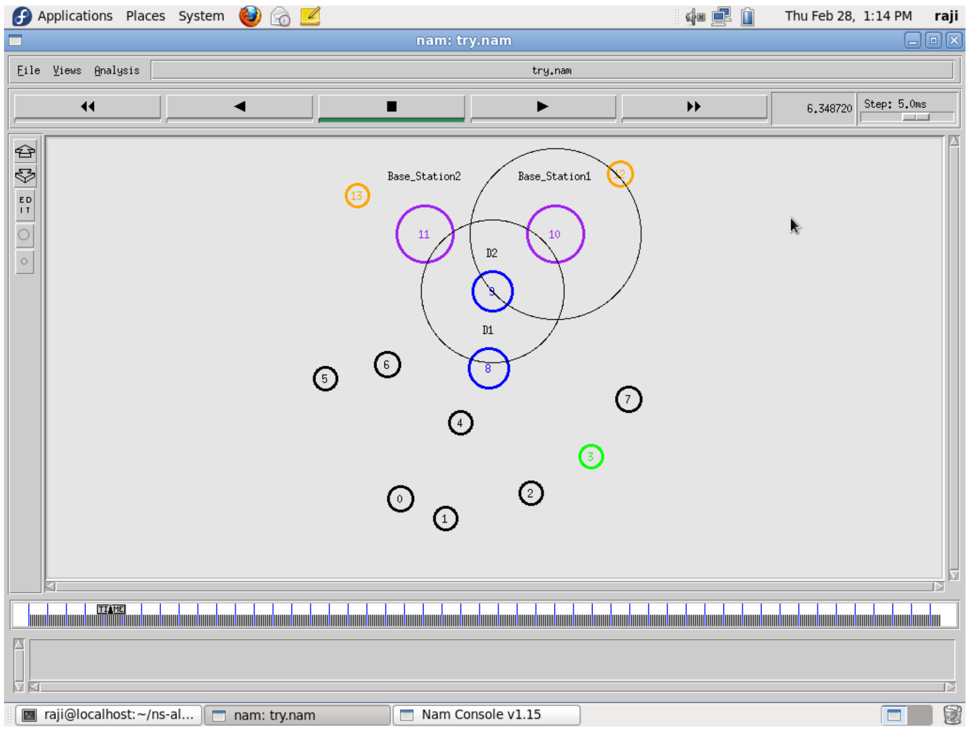
9.1.1 creation of nodes



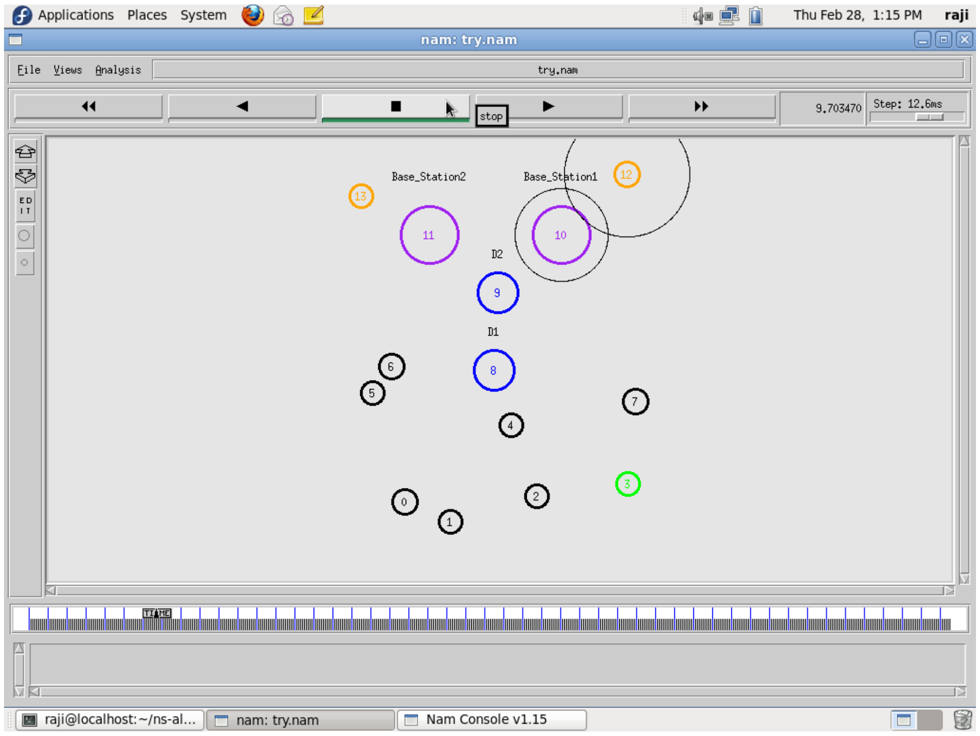
9.1.2 Node communicating with nova device1



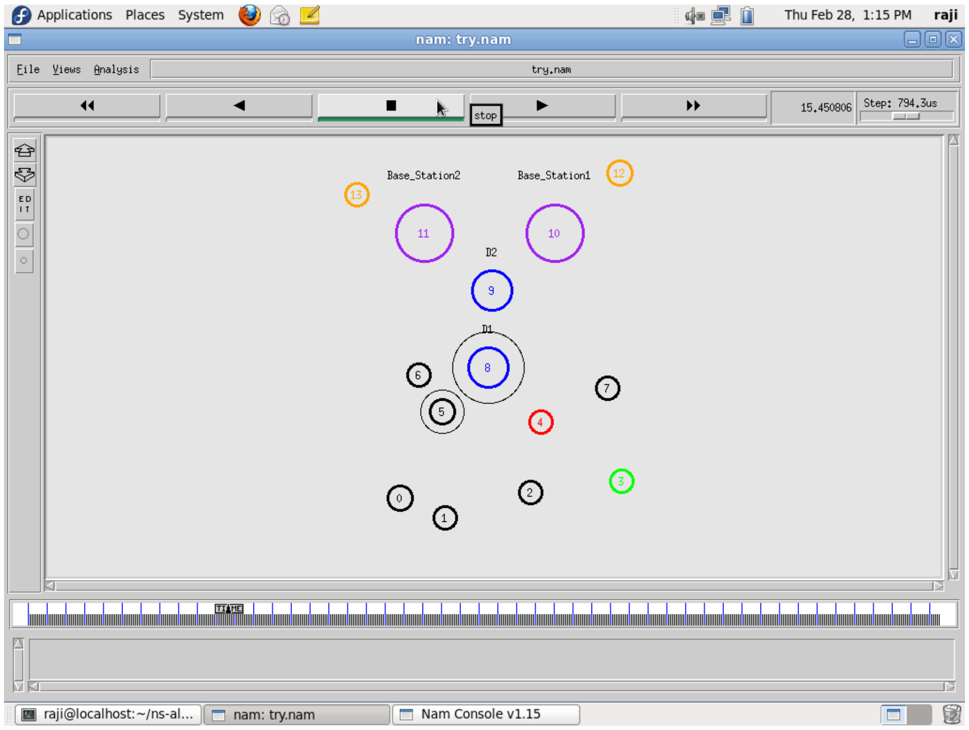
9.1.3 Node classified as authorized



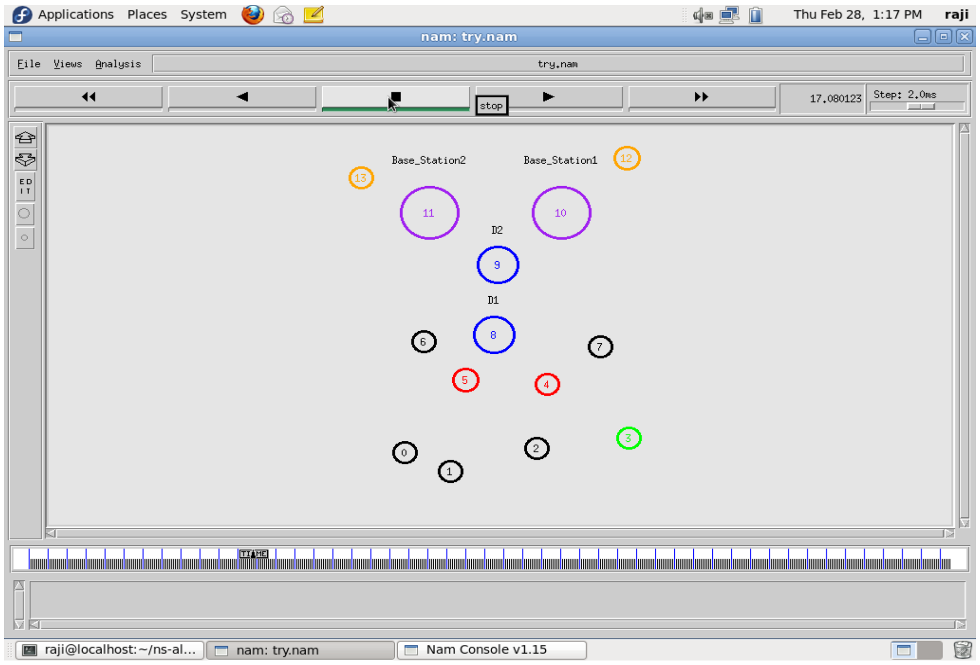
9.1.4 Communication between the nova device2 and the base station



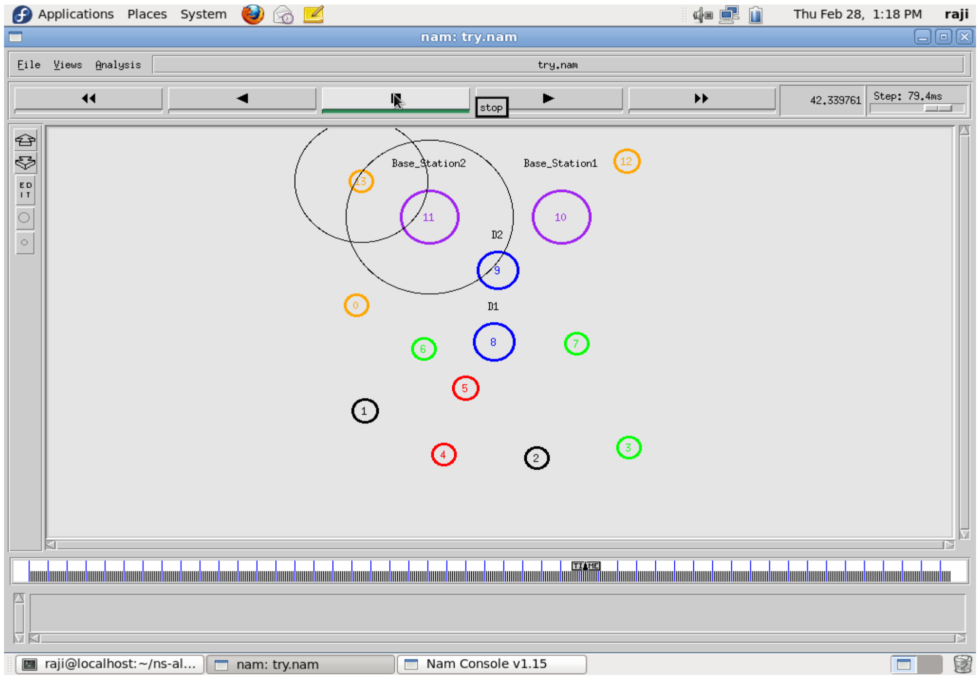
9.1.5 Communication between base station and the mobile node



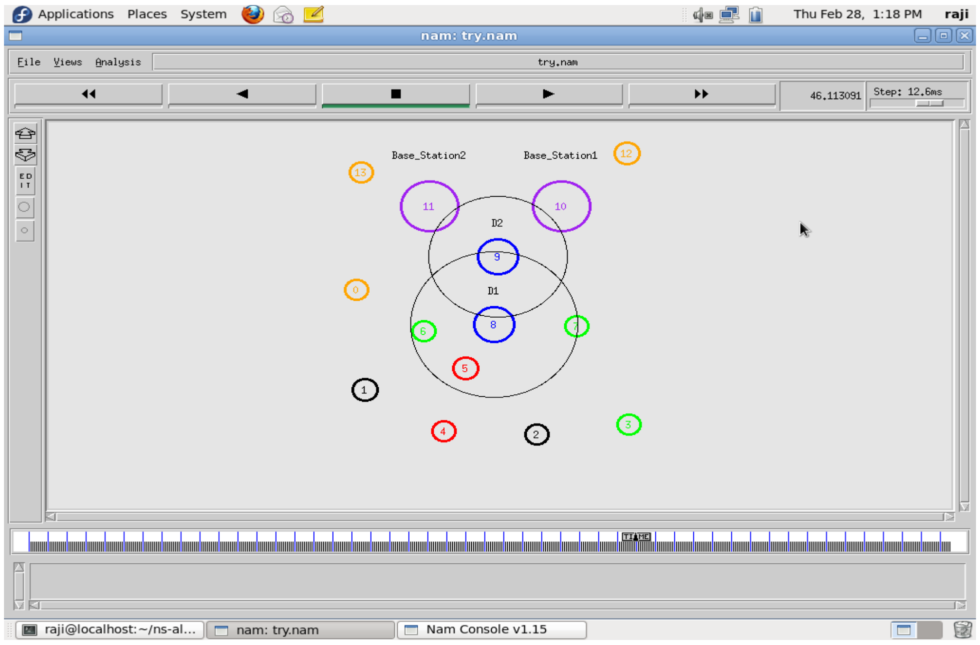
9.1.6 Node 4 classified as unauthorized



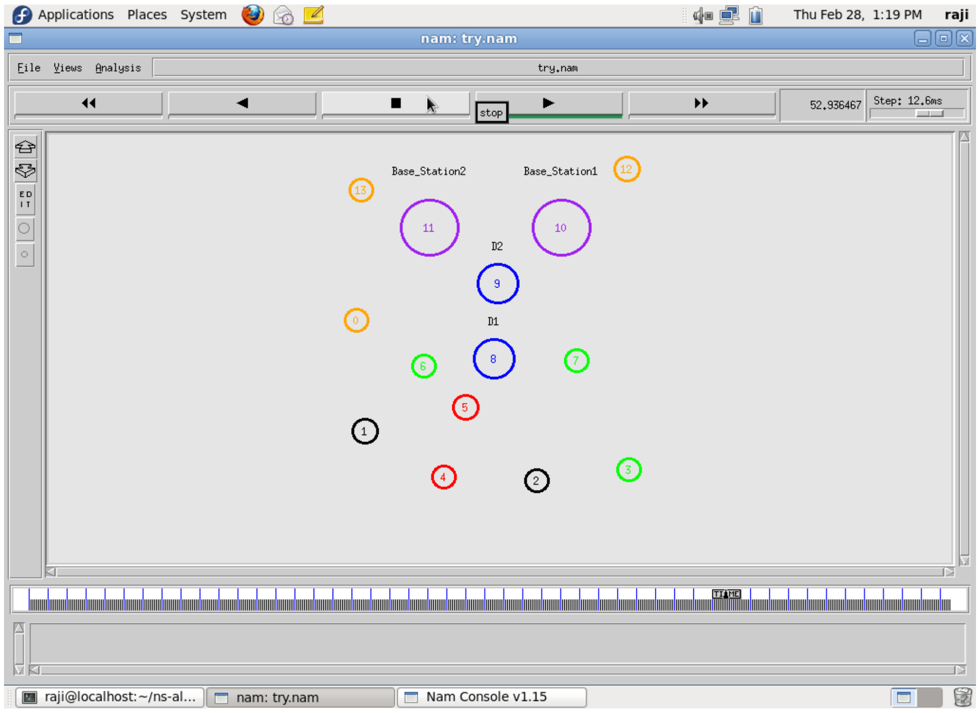
9.1.7 Node 5 classified as unauthorized



9.1.8 Incoming call



9.1.9 Communication between nova device2 and device1



9.1.10 After all communications

CHAPTER X
REFERENCES

[1]. An Introduction to NS, Nam and OTcl scripting, Paul Meeneghan and Declan Delaney, NUIM-CS-TR-2004-05.

[2]. A Software Testing Primer, An Introduction to Software Testing by Nick Jenkins 2008

[3]. Introduction to Network Simulator by Teerawat Issariyakul and Ekram Hossain-NS22009 Springer Science Business Media, LLC

[4]. Jochen Schiller, "Mobile Communications" Second Edition, PearsonEducation,*2003*

.

[5]. Mark Gresis's Tutorial for the UCB/LBNL/VINT Network Simulator "ns", http://www.isi.edu/nsnam/ns/tutorial/index.html

[6]. Review of the Mobile Phone Jammer Prohibition, Public Discussion Paper,2010, Published by the Australian Communications and Media Authority.

[7]. The ns Manual (formerly ns Notes and Documentation), The VINT Project Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC. Kevin Fall hkfall@ee.lbl.govi, Editor Kannan Varadhan, hkannan@catarina.usc.edui, Editor July 25, 2008.