Digital Signal Processing Extra-tropical Cyclones Warning System using WiMAX

Mohamed Ahmed Al-Breiki

School of Engineering and Technology

UNIVERSITY OF HERTFORDSHIRE

Thesis submitted to the University of Hertfordshire in partial

fulfilment of the requirements

of the Degree of Doctor of Philosophy

January 2013

Abstract

This research project proposed a unique solution to make use of these base stations to keep all subscribers alerted with warning of possible disaster should that be required. As the current, network does not provide a provision for such a noble approach, a new network model has been developed and simulated to interface a sensor (weather station, WeS), with WiMAX weather station. The weather station is based on DSP processor to receive a digitised sensor values, process these values, analyse them and if they fall within the alert zones, packet them according to WiMAX protocol and send them to subscribers. The developed standard bypasses any commercial network to offer free transmission to subscribers. This setup is also able to extract information on weather condition or react on uncertainty, i.e. disaster scenarios.

Natural disasters, such as torrent, tornado/ hurricane, volcano eruption, earthquake, Tsunamis or landslide are increasing. Unfortunately they bring with them human tragedies, environment catastrophes, villages, cities and counties are subject to endless devastation during and after the destructive forces.

Water, electricity and gas supply are most disrupted and difficult to restore in short time. However, communication is another item that can be affected adversely but WLAN with specific considerations, should be excluded from the effect. This project presents a solution, albeit minor relative to the maximum effect of the disaster, but will keep the telecommunication/communication in operation. Our novel technique, a “Clone Wireless Wide Area Network (CloneWAN)” is a clone wireless network to the wired Network. In the event of natural calamities, it gives continuity of network operation. It is based on WiMAX.

The realization of CloneWAN has been formed and simulated to set the national network of the UAE at its correct form.

CloneWAN model has been simulated with Opnet platform. All results revealed that the model is complete. The interface to Alerting System is discussed. Results show that the dynamic behavior of the parameters delay and Throughput of CloneWAN model is stable over various and different load scenarios.

WiMAX is a de-facto standard in the current and future network requirement standards. Its main component is the Base Station which is normally stationed in the air, high enough to

couple signals from other base stations. It is purpose is merely focused on networking signals for commercial purposes.

The suggested hardware interface for the Weather Station is based on DSP SHARC processor. The model has been written in C and simulated under Opnet package. A number of scenarios have been set to represent different disasters worldwide. All results are listed and discussed later in the thesis.

Acknowledgment

الحمد لله (Al-Hamodo Lillah: Praise to God) for getting this project to this stage. لله (Allah) Almighty intention and blessings steered me to set my mind to start on further study, to travel to the UK, to think solely for the project, read/ research and study at hourly rate, experiment at daily rate, question and ask for help from the University friends and strangers to progress and cross so many stumbling blocks, a truly holy journey.

I owe everything to my honourable father for all his support and guidance throughout my life, especially within the last 4 years.

I would sincerely like to thank my dearest family for their support and patients as they have been in my thoughts so many short and long nights I have endured being so close and yet so far from them.

I would like to thank the Deputy Prime Minister and the Minister of Interior of the UAE, His Highness Lieutenant General Sheikh Saif Bin Zayed Al Nahyan for his direct and indirect care, directions and to give us this unique opportunity and lifetime chance to gain further knowledge and experience in different fields of study and research, a very thoughtful vision for a better UAE nation.

To the Secretary-General of the Minister of Interior of the UAE, General Nasser Salem Al Khreibani Al Neimi, my gratefulness for the advice to select this route of research, it is undoubtedly the state of the art technology that would benefit mankind with little warning from the current extreme and rapid changes in the weather.

From day one of the arrival to the UK to the University of Hertfordshire, the day my research started, I was fully, wisely and comprehensively supervised by Professor T. Alukaidey, my sincere gratitude to him.

Mr Johann Siau has complemented the supervision with the right advice and suggestions to move this project forward, I am very grateful to him.

Two wonderful ladies, from the research office of the University of Hertfordshire, School of Engineering and Technology, Mrs Lorraine Nicholls and Mrs Avis Cowley, many thanks for your professional and understanding to my constant requirements.

List of Figures

Abbreviation

ACE	Application Characterization Environment
AMC	Adaptive Modulation and Coding
BGP	Border Gateway Protocol
BS	Base Station
CAP	Common Alerting Protocol
CloneWAN	Clone Wireless Wide Area Network
CN	Community Network
CPE	Customer Premise Equipment
DL	Downlink
ESD	External System Domain.
FDD	Frequency Division Duplexing
FFTs	Fast Fourier Transforms
FSMs	Finite State Machines
ICI	Interface Control Information
IGRP	Interior Gateway Routing Protocol
IPTV	Internet Protocol Television
KP	Kernel Process
LOS	Line Of Sight
LTE	Long-Term Evolution
MS	Mobile Station
OFDM	Orthogonal Frequency Division Multiplexing
QoS	Quality of Service
PtM	Point-to-Multipoint

PtP	Point-to-Point
SDU	Session Data Unit
SITL	System In The Loop
SS	Subscriber Station
VoIP	Voice over Internet protocol
UDP	User Datagram Protocol
UL	Upperlink
TCP	Transmission Control Protocol
TDD	Time Division Duplexing
WAN	Wide Area Network
WeS	Weather Station
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Networks
WPAN	Wireless Personal Area Network

Table of Contents

Chapter one: Introduction

1.1 Overview

This research is aimed to provide a network setup that in case of natural disasters, such as flood, tornado/hurricane, volcano eruption, earthquake, Tsunamis or landslide, the public would still be able communicating as usual as possible.

Section 1.2 presents a survey to four wireless network that is provided by private research institutes or industry. These four networks have been assessed against their advantages and disadvantages in section 1.3. A fifth new method suggested in section 1.4 called CloneWAN model which is based on WiMAX standard. CloneWAN is the hub for sending alerts to subscribers by using WiMAX base stations. The alert standard, CAP, is presented in section 1.5. Further details on CloneWAN with CAP are described in section 1.6.

1.2 Survey on WLANs for Disaster Recovery Operations

Four approaches have been surveyed and reported here. These reports were published on completed and tried projects in real life. Some of them have had large funds to support them and some took more than 5 years to achieve the results. The discussion on these reports is based on critical points. For example, do these networks provide services free to subscribers? Can the network be implemented anywhere in the world? Can it be used for normal operation and extreme weather condition alerts? The end of this session will realise the answer and a suggestion to new network is presented.

1.2.1 A Network on Wheels for Emergency Response and Disaster Recovery Operations

The 911-network on wheels (Abusch-Magder et al., 2007) solution is a novel portable cellular system based on base station routers (BSRs) that does not require any pre-existing wireless infrastructure and provides capacity and coverage on demand. It is an auto-configurable system with a fully integrated service architecture that can be deployed as a single-cell solution for local communication or be configured to operate as an ad hoc network of cells. Employing commercial technology for emergency response networks provides significant benefits:

- Wide availability of commercial handsets during emergencies
- Significant cost savings from economies of scale because of large-scale deployment of commercial technologies

- Rapid evolution and feature development in handset capabilities and services driven by competition in the commercial market
- Multi-vendor interoperable solutions.

Figure 1.2.1.1 shows the architecture of 911-NOW.

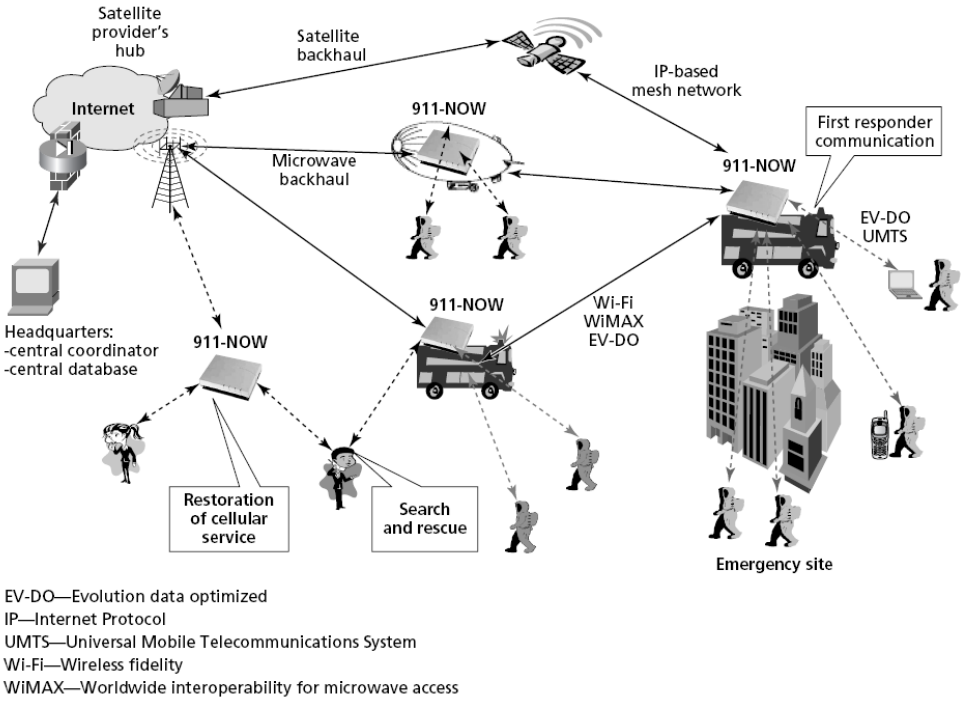


Figure 1.2.1.1: The Full Topology of 911-NOW (Abusch-Magder et al., 2007)

However, one disadvantage about 911-NOW, it assumes that there are no emergency cases.

1.2.2 A Solar-Powered WiMAX Base Station

A solar-powered WiMAX base station proposed by Intel (Application Note, 2006) can provide a particularly convenient and necessary solution where areas with little or no access to electric power. During disaster recovery operations are often conducted where electric power is unavailable from the grid, and WiMAX has proven valuable in several recovery missions.

This model is meant for fixed WLAN or mobile WLAN. As the emphasis is on Solar cell, no indication if it will continue to operate in areas mostly clouded or there is no sun. Moreover, it lacks the ability to extract information on weather condition or react on uncertainty, i.e. disaster scenarios.

1.2.3 Field Test Report WiMAX Frequency Sharing with FSS Earth Stations

Field Test Report WiMAX Frequency Sharing with FSS Earth Stations has brought about very interesting outcome to WLAN applications (Ames, 2008). The portion of C-band (frequency band 3.4– 4.2 GHz) is designated by several national administrations around the world for use by terrestrial wireless applications such as WiMAX and future mobile services. The C-band is already in use by satellite services, radar systems and domestic microwave links. C-band services cover large areas, facilitate intercontinental and global communications and provide a wide range of services in developing countries. The C-band provides a robust, reliable platform for such critical applications as distance learning, telemedicine, universal access, disaster recovery, and television transmission in many remote and tropical regions. It has proven to be exceptionally useful for disaster recovery in tropical areas because the C-band covers wide areas with minimum susceptibility to rain fade.

Among other problems, during the operation, testing showed that the transmit signal could cause significant problems to a digital signal in access of 12 km away (Ames, 2008).

1.2.4 WLAN in Disaster and Emergency Response (WIDER)

To provide an enhanced communications network in disaster sites WIDER project was initiated by Ericsson Response and KTH (ERRICSON, 2003). This project gained attention by international telecommunications and humanitarian organizations (ITU, Red Cross and others). The motivation behind the project was increased size and frequency of Humanitarian Relief Organization, inefficient contemporary communication (mainly voice), lack of data communication, waste of resources by using satellite communication even for local calls, need for internet access and related services.

The main aspects of WIDER are as follows:

- A wireless network that connects relief organization camps allowing local communication.
- A standalone network that operates in areas when there is little or no infrastructure.
- A highly portable Internet Service Provider network that uses a satellite system to reach Internet.

WIDER is not intended to provide the messaging service freely to subscribers, it is managed by a network organisation, by Ericsson.

1.3 Comparison to the Above Four Operations

Advantages and disadvantages for the above four network methods are listed here:

- 911-NOW solution is novel but doesn't rely on infrastructure and does not communicate with current LAN or WLAN. It assumes there are no emergency cases.
- Intel suggested a solar-powered WLAN setup. However, there is no indication it would continue operate where there is limited or no sun plus no consideration have been taken to the disaster situations.
- Among other problems for the third method, during the operation, testing showed that the transmit signal could cause significant problems to a digital signal in access of 12 km away.
- On the other hand, WIDER by Ericsson has enhanced communications network in disaster sites. WIDER is a standalone setup which means it has to replace the current WLAN whether a disaster struck or not.

Hence, the search is still on for better alternatives. Chapter 3 is suggesting a novel approach to WLAN that will clone the current setup in case of emergencies but based on new network set as described in Chapter 4, that takes weather information, react accordingly and send an alert in case of extreme conditions.

1.4 CloneWAN Hardware Model

The United Arab Emirates (UAE) have been used as a case study for the new WAN architecture. Figure 1.1.1 shows CloneWAN architecture for UAE case study.

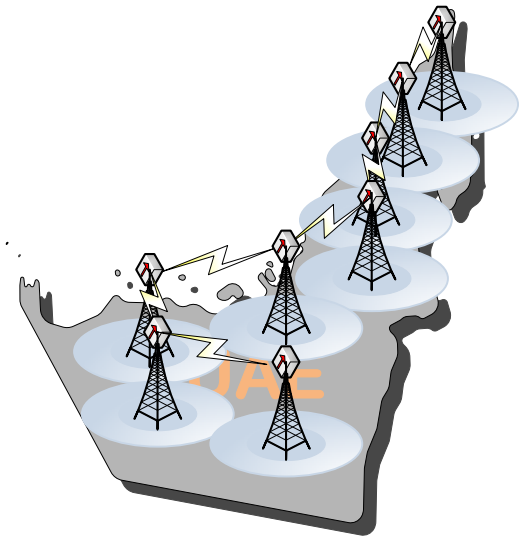


Figure 1.4.1: UAE Proposed CloneWAN Base Stations Plan

The suggested system was based on WLAN. However, the latest development on WiMAX gives an excellent alternative (Andrews et al, 2007). One of the main alternative is the bandwidth of WiMAX is superior to WLAN. Another alternative is the reception of video on motion which reported to be over 70 Mph. Hence, designing, developing and testing a WiMAX model with Alert facilities is the aim of this research program.

As WiMAX Base Station transmits within a specific radius and taking into consideration the area of Abu Dhabi the capital of the UA, it is suggested that the system is to be based on 7 base stations interlinked and interoperable to provide full facility at the scale of the UAE country to cover and provide all sector of public and customers' requirements. The weather forecast sensor to be positioned within the base stations.

Section 1.4 explains all types of network topologies. A brief on the architecture of WiMAX is provided in section 1.4.1. Section 1.5 reveals the details on the Alerting System: Common Alerting Protocol Version 1.2. CloneWAN with WiMAX as the base unit interfaced with the Alerting System is proposed in section 1.6. Guidance to the thesis reported in section 1.7.

1.4.1 Network Types and WiMAX

Network types (Nuaymi, 2007) have been categorized in WAN, WMAN, WLAN and WPAN. The illustration shown in figure 1.4.1.1 is relative to the distance coverage.

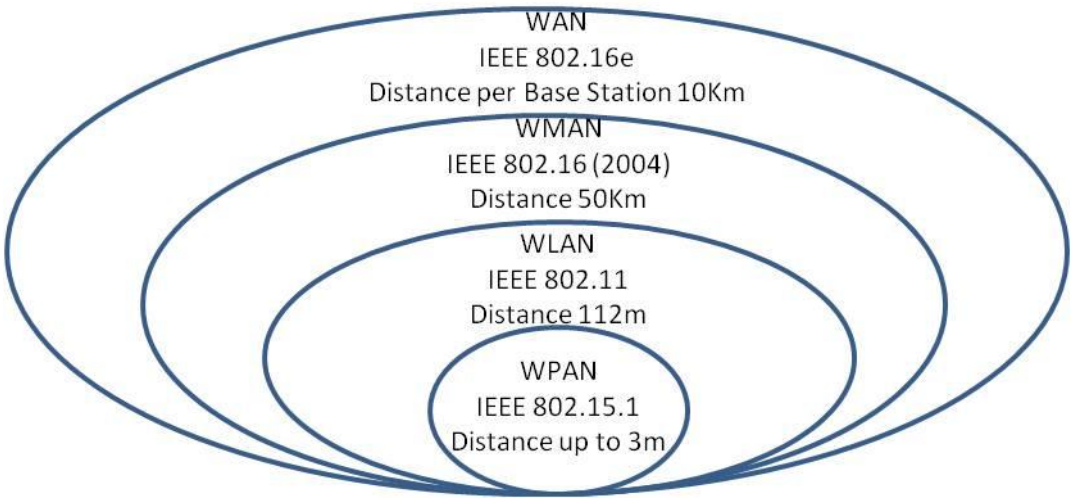


Figure: 1.4.1.1: Network Types

Example of WPAN is Bluetooth, WLAN is WiFi, and WAN is WiMAX.

1.4.1.1 WiMAX

WiMAX is Worldwide Interoperability for Microwave Access. WiMAX is approved as IEEE standard and designated as 802.16-2004 (fixed wireless applications) and 802.16e-2005

(mobile wireless). WiMAX could be considered as a Community Network (CN) (Andrews et al, 2007).

Current telecommunications infrastructures provide services via:

- Broadband Internet Access
- Landline Telephone
- Cable or Satellite TV and
- Mobile Data and Mobile (Cell) phone

The hub to the above services is potentially measured by WiMAX. For example, WiMAX fixed wireless configuration can replace the landline telephones and TV coaxial cables. At the same time, fixed wireless can provide ISP services. Mobile phones and mobile data by default are provided by WiMAX mobile wireless configurations.

1.4.1.2 Fixed WiMAX

It offers point-to-point (PtP) and point-to-multipoint (PtM) solutions as shown in figure 1.4.1.2.1.

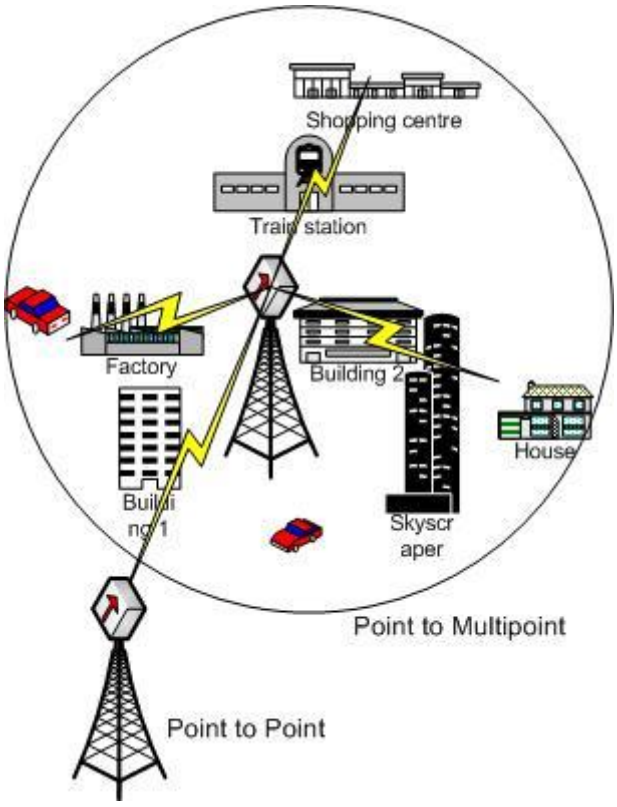


Figure 1.4.1.2.1: WiMAX PtP and PtM.

WiMAX can provide two forms of wireless service:

No-line-of-sight: With a small antenna on the receiver end, for example a PC, it will connect to the radio WiMAX base station. In this mode, WiMAX uses a lower frequency range - 2 GHz to 11 GHz.

Line-of-sight: As shown in figure 1.4.1.2.1, a fixed dish antenna points straight at the WiMAX tower from a tower (or similar setup). The line-of-sight connection is stronger and more stable to send data more reliably with fewer errors (Nuaymi, 2007). Line-of-sight transmissions use higher frequencies, with ranges reaching a possible 66 GHz (Nuaymi, 2007).

Non-line-of sight services from a base station to a subscriber station within 10Km to deliver 40 Mbps are part of WiMAX provision:

- Voice over Internet protocol (VoIP) as telephone company substitute.
- Internet Protocol Television (IPTV) as cable TV substitute.
- Backhaul for Wi-Fi hotspots and cell phone towers.
- Mobile telephone service.
- Mobile data TV.
- Mobile emergency response services.
- Wireless backhaul as substitute for fibre optic cable.

1.4.1.3 Mobile WiMAX

It enables streaming video and other mobile (cell) phone applications to vehicles at over 70Mph (Nuaymi, 2007). Its antennas' specifications allow it to penetrate buildings and offers improved security measures over fixed WiMAX (Nuaymi, 2007). It is already suggested that it is valuable for emerging services such as mobile TV and gaming. Hence, it is a perfect domain for CloneWAN requirements.

1.4.1.4 WiMAX Architecture

The main components that form WiMAX system are:

a) Transmitter - Base Station.

The Base Station (BS) is a transmitter and an antenna fitted on top of a roof, building or tower.

BSs would use the MAC layer defined in the standard to offer a common interface that makes the networks interoperable and would allocate uplink and downlink bandwidth to subscribers according to their needs, on an essentially real-time basis.

b) Receiver – MODEM.

WiMAX MODEM receiver could be a standalone box to be interfaced to a laptop or PC or any other system. WiMAX modem may have a separate antenna. This is also referred as customer premise equipment (CPE).

c) Backhaul – Internet Cloud.

Internet Cloud connects bi-directionally a WiMAX tower station to a high-bandwidth, wired connection.

It is possible to connect several base stations to one another using high-speed backhaul microwave line-of-site links. This would also allow for roaming by a WiMAX subscriber from one base station coverage area to another, similar to the roaming enabled by cell phones.

WiMAX architecture with two cells could be visualized as shown in figure 1.4.1.4.1.

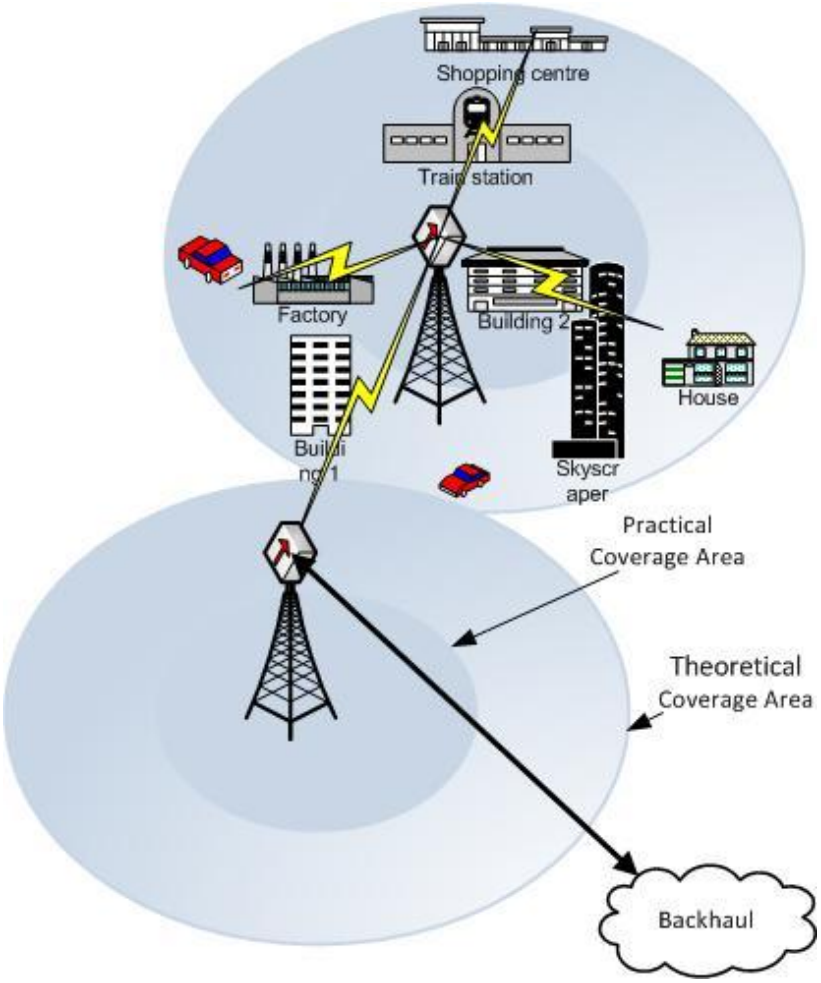


Figure 1.4.1.4.1: Two Cells WiMAX Architecture.

Without interference, the coverage of each radio WiMAX BS is described theoretically as 50 km and practically as 10 km. The two cells shown in figure 1.4.1.4.1 illustrate outer and inner cells to represents the theoretical and practical coverage respectively. Note that, WiMAX practical coverage is over 30 times wider than the theoretical WiFi coverage.

1.4.1.5 WiMAX Features

WiMAX features (IEEE, 2005) are listed below, modelled and incorporated in CloneWAN architecture:

- OFDM-based physical layer.
- Very high peak data rates.
- Scalable bandwidth and data rate support.
- Adaptive modulation and coding (AMC).
- Link-layer retransmission.

- Support for TDD and FDD.
- Orthogonal frequency division multiple access (OFDMA).
- Flexible and dynamic per user resources allocation.
- Support for advanced antenna techniques.
- Quality-of-service support.
- Support for mobility.
- IP-based architecture.

The above WiMAX features have been simulated within CloneWAN Opnet model as described in Chapter 3.

1.4.1.6 CloneWAN Weather Station

The Weather Station (WeS) is based on set of sensors that are interfaced to a processor via A to D converter. These samples should be processed and Digital Signal Processor (DSP) has been selected to execute simple weather prediction mathematical formulas. The DSP platform hardware will be interfaced to WiMAX base Station. Software code will be written to collect the weather information and provide it to WiMAX. The full hardware design of WeS is described in Chapter 3.

1.5 Alert System

CloneWAN disseminated warning messages; hence it incorporates a dissemination system. This system covers all possible cases of natural hazard warnings. The weather forecast sensing system is an input interface to CloneWAN. It would be practical to implement a standard to disseminate the warring messages. The Common Alerting Protocol (CAP) (CAP, 2010) provides an open, non-proprietary digital message format for all types of alerts and notifications.

1.5.1 CAP capabilities

The common Alerting Protocol (CAP) is intended to standardise the warning messages to subscribers. It is an XML based messages that the user can read, hear or view the alerts. There are large number of situation were the users should receive alerts specially if the weather conditions are extreme. The list of CAP application capabilities is:

- Flexible geographic targeting using latitude/longitude shapes and other geospatial representations in three dimensions. Hence it specifies any location in the glob.
- Multilingual and multi-audience messaging. This is a handy feature as it will describe the alert in Arabic language which needed in the UAE.
- Phased and delayed effective times and expirations. This feature is useful in case if the Met office predicts in advance harsh weather events. The DSP would workout the timing and sent the Alerts to SSs.
- Enhanced message update and cancellation features. This feature again related to predication scenarios as mentioned above feature, as some of the predictions get corrected and hence the SSs should be informed.
- Template support for framing complete and effective warning messages. There are few templates and the ones that are related to temperatures and illuminations will be used.
- Compatible with digital signature capability. This feature is a measure of security.
- Facility for images and audio. Should WeS implemented and matured, images and sound can be sent to SSs. Sending images will enforce the alert message but may help the deaf. Similarly, the sound will enforce the alert message but may help blinds.

One of the key benefits of CAP message format that it can be converted to and from the "native" formats of all kinds of sensor and alerting technologies, forming a basis for a technology-independent national and international "warning internet.

1.5.2 Alert Message Model

There are four segments that form CAP Alert messages;

- **<alert>**
- **<info>**
- **<resource>**
- **<area>**

The above four segments with the CAP features listed in section 1.5.1 generates any weather scenario WeS may encounter.

1.5.3 CAP Message Structure

CAP message structure is described below. Since Opnet is based on C++, the XML code could be compiled on C++ to be integrated with CloneWAN Opnet model.

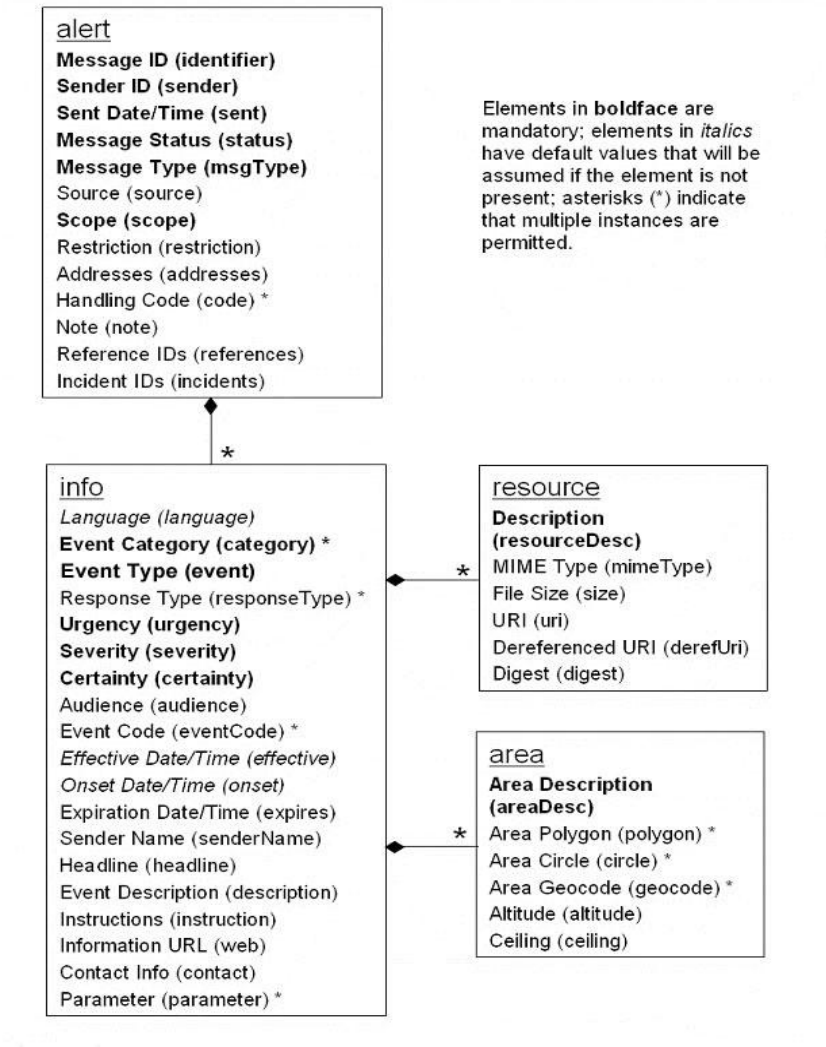


Figure 1.5.3.1: CAP Message Structure.

The <alert> segment provides basic information about the current message: its purpose, its source and its status, as well as a unique identifier for the current message and links to any other, related messages. An <alert> segment may be used alone for message acknowledgements, cancellations or other system functions, but most <alert> segments will include at least one <info> segment.

The <info> segment describes an anticipated or actual event in terms of its urgency (time available to prepare), severity (intensity of impact) and certainty (confidence in the observation or prediction), as well as providing both categorical and textual descriptions of the subject event. It may also provide instructions for appropriate response by message recipients and various other details (hazard duration, technical parameters, contact information, links to additional information sources, etc.) Multiple <info> segments may be used to describe differing parameters (e.g., for different probability or intensity “bands”) or to provide the information in multiple languages. Each <info> segment may include one or more <area> and/or <resource> segments.

The <resource> segment provides an optional reference to additional information related to the <info> segment within which it appears in the form of a digital asset such as an image or audio file.

The <area> segment describes a geographic area to which the <info> segment in which it appears applies. Textual and coded descriptions (such as postal codes) are supported, but the preferred representations use geospatial shapes (polygons and circles) and an altitude or altitude range, expressed in standard latitude / longitude / altitude terms in accordance with a specified geospatial datum.

1.6 WiMAX and the Alerting System for CloneWAN

The proposed CloneWAN software model is drawn as shown here in figure 1.6.1.

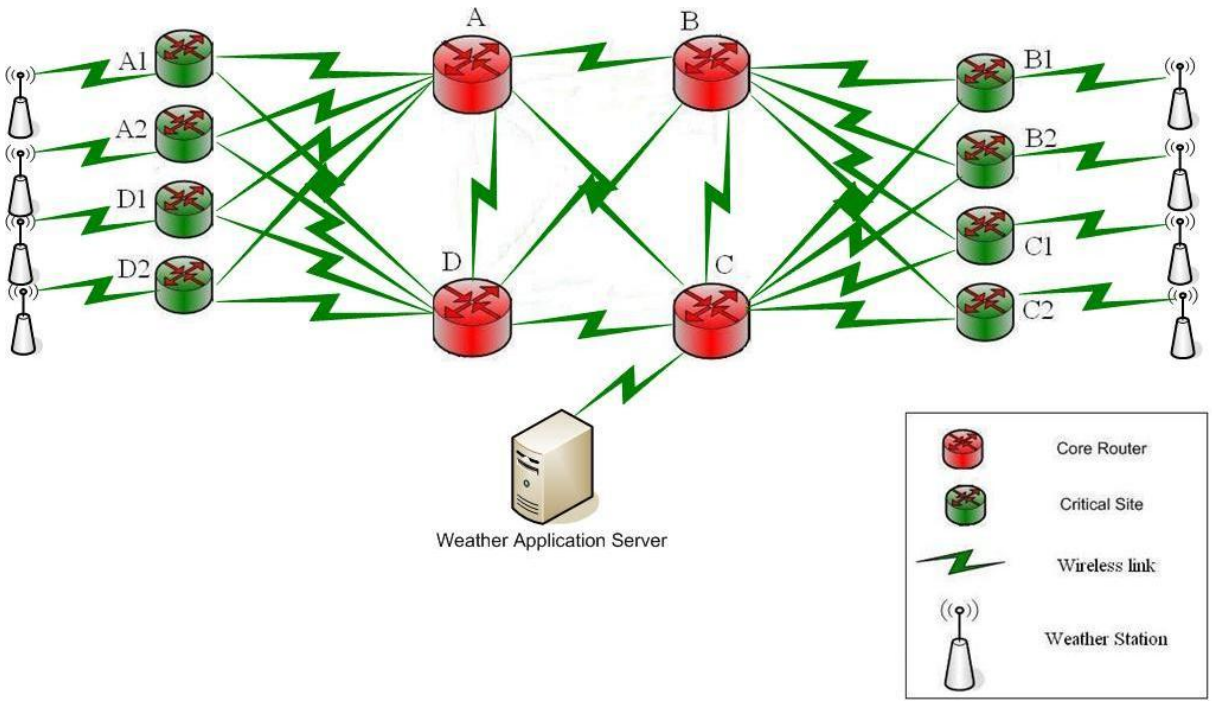


Figure 1.6.1: Proposed CloneWAN Software Model

The above proposal includes set of routers, weather sensors and application server. This model will be simulated using Opnet in chapter 3.

1.7 Research Objectives

The list of objectives for this research project is:

- Design a complete network that suits the UAE
- To be simulated with Opnet
- Design the hardware interface of a Weather Station (WeS)
- Develop the required code for WeS
- To be simulated

1.8 Guidance to the Thesis

Chapter 2 provides a background details on Opnet. The understanding of the hierarchical structure of Opnet helps to follow the model work of the Weather Station (WeS) developed in Chapter 4.

CloneWAN architecture is presented fully in Chapter 3. Here, the architecture of CloneWAN has been discussed and the results to the current model of CloneWAN are included. This architecture is new and proven to work as demonstrated in this chapter.

The following chapter, Chapter 4 provides full design, coding and simulation to Weather Station (WeS). The work of developing WeS model is based on using C/C++ and Proto C languages. Opnet provided a set of functions that enable the code of the model to be statistically analysed and revealed. The transmitted and received data have packet-based on the available current network standard on Opnet. The work here represents a novel approach to develop WeS model for WiMAX.

Chapter 5 presents the results of WeS model. The model is based on Opnet environment and hence the snapshots of Opnet performance have been listed and discussed in this chapter.

The conclusion chapter, Chapter 6 listed the salient points of this project report. In addition, a section on future work has been included here to guide future researchers to the next step of the current research.

Chapter two: Simulation Tools

Opnet (Optimum network) is a network simulator that encompasses wide range of network components. It is able to simulate a number of scenarios for one project application (Opnet 17.1, 2012). Other network simulators such as Ns or Omnet lack the simplicity and visual ability to model a protocol such as WiMAX. Hence, Opnet was the simulator used for this project.

CloneWAN adapts Transmission Control Protocol (TCP). TCP is a reliable protocol in the IP-suite. Opnet meets some requirements to simulate and model IP-suite applications:

- TCP's slow start characteristics: This is a known feature for TCP and Opnet does offer low bit rate at the start of the simulation session
- Link breakdown and link restart: Opnet able to simulate routing protocols under different circumstances
- QoS: Opnet able to examine how Quality of Service (QoS) affects transfers in packet coupled (IP) and virtual circuit coupled (ATM) networks

This Chapter presents a brief detail on Opnet in section 2.1. Opnet main components are called Network Model, Node Model and Process Model. Section 2.2 discusses these components. Section 2.2 continue to cover Opnet sub-cores etc. up to the ones that is called States where C or C++ has to be provided to an interface to the new hardware items, new algorithms or new protocols. Section 2.3 is related to the output of Opnet in single or multiple formats. An example on Wireless model has been provided in section 2.4. This chapter ends with a summary in section 2.5.

2.1 Introduction to Opnet

Opnet is an environment that allows modelling and simulating various types of network topology and technology following top down description.

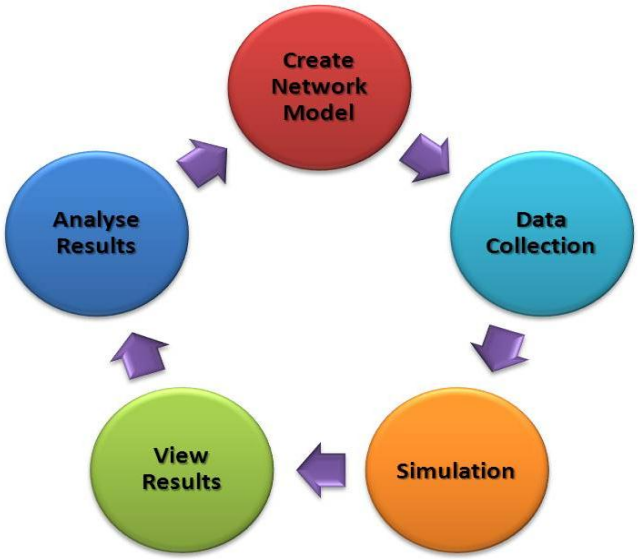


Figure 2.1.1: Opnet Workflow Cycle

The network model is created with a specific set of details on Opnet environment as shown in figure 2.1.1. The data collection and simulation are straightforward options. View and analyse the results are the final stages of the cycle. The cycle shown in figure 2.1.1 is repeated up until the specifications have been met.

2.2 The main components of Opnet

Opnet Workflow is composed out of the following main components:

The Project Editor is the main staging area to create a network simulation. It allows developing:

- Network Objects - Subnets

Network topology requires one or more subnets. The network in figure 2.2.1 is an example of LAN Network Model using Opnet platform.

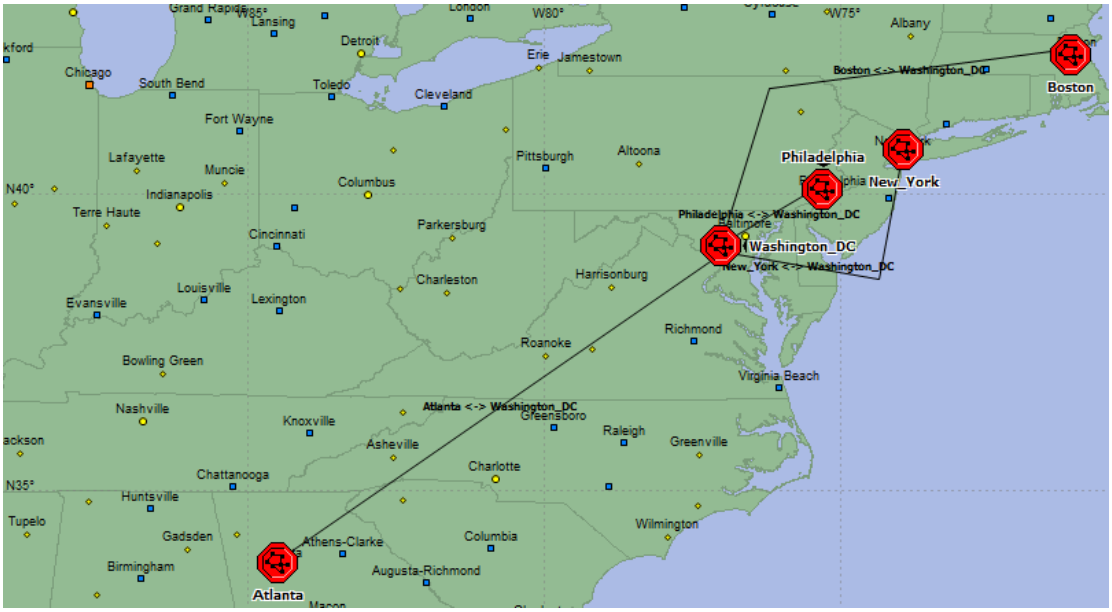


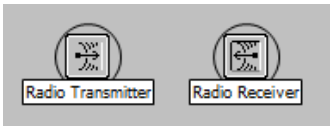
Figure 2.2.1: Network Model

The network model links between Atlanta, Washington DC, Philadelphia, New York and Boston as specified by the 5 fixed subnetwork, the red nodes. 5 of the nodes are point-to-point linked.

- Network Objects - Link

Link objects model physical layer effects between nodes, such as delays, noise, etc. Examples of links are point-to-point link, bus link and radio link:

- **A point-to-point** link transfers data between two fixed nodes
- **A bus link** transfers data among many nodes and is a shared media
- **A radio link**, established during a simulation, can be created between any radio transmitter-receiver channel pair. Satellite and mobile nodes must use radio links. Fixed nodes may use radio links. A radio link is not drawn but is established if nodes contain radio transceivers

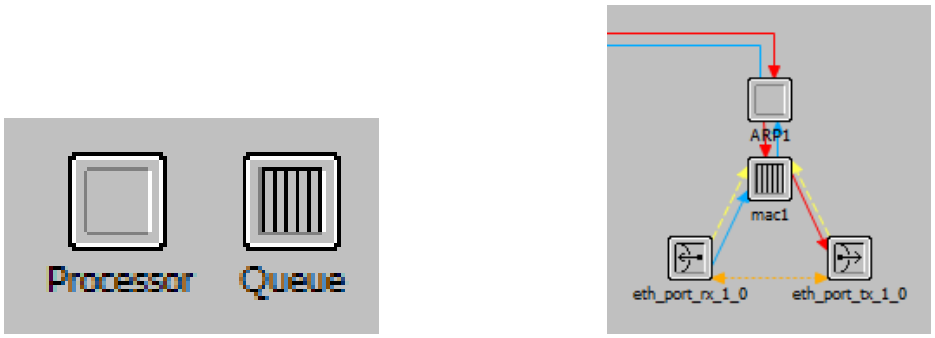


- Node Objects - Modules

Modules are the basic building blocks of node models. Modules include processors, queues, transceivers, and generators:

- Processors are the primary general purpose building blocks of node models, and are fully programmable
- Queues offer all the functionality of processors, and can also buffer and manage a collection of data packets.

Processor and Queue modules examples are shown in figure 2.2.2.



(a) (b)

Figure 2.2.2: (a) – Processor & Queue Module Examples, (b) – Arrival Processor (ARP1) is linked to Queue mac1 in Boston Subnet

Figure 2.2.3 shows all required layers of Boston Subnet.

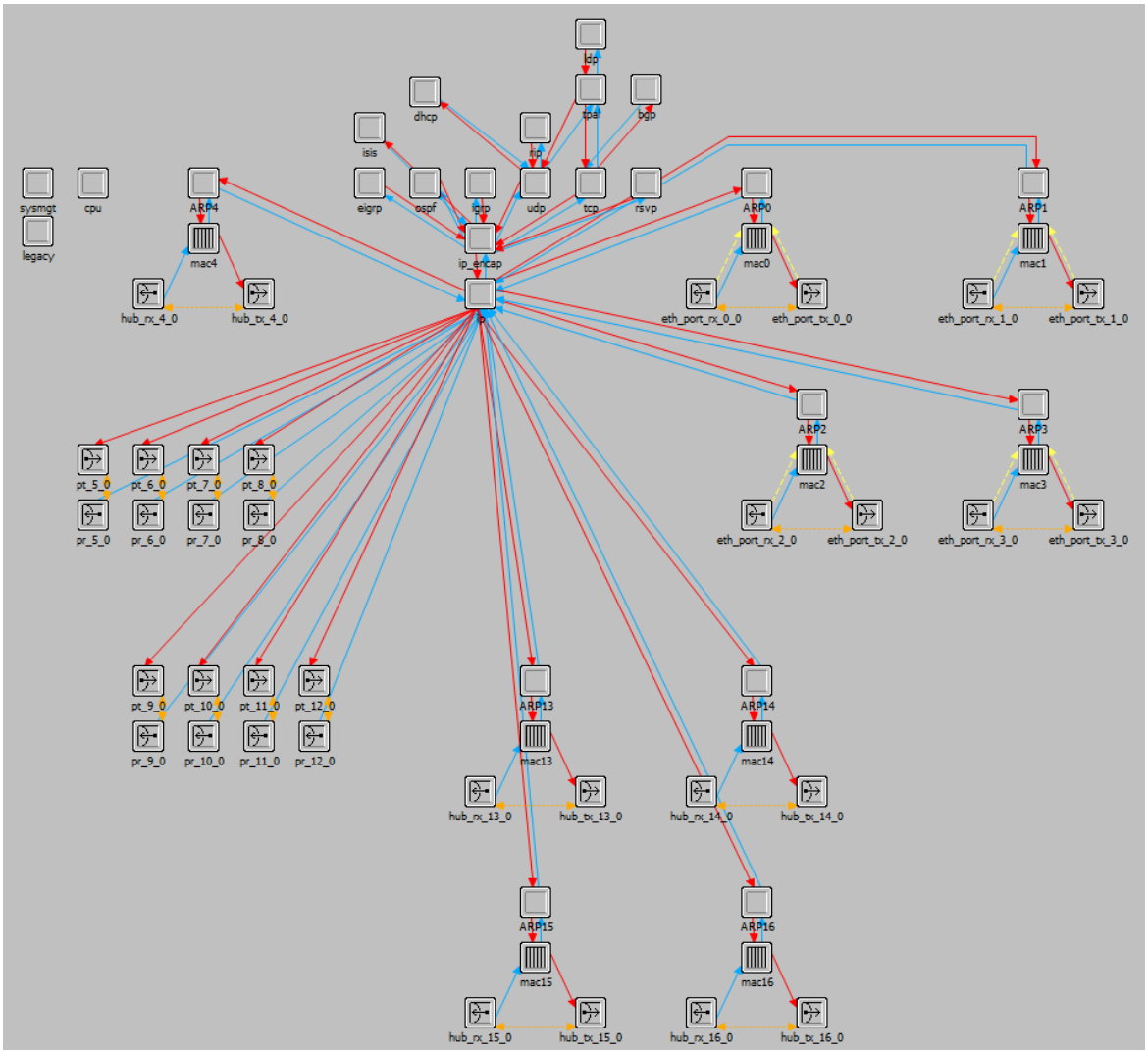


Figure 2.2.3: Router Processor Node

- Process Model - States

C/C++ code resides/embedded within States. There are two hot spots per state. This is shown in figure 2.2.4.

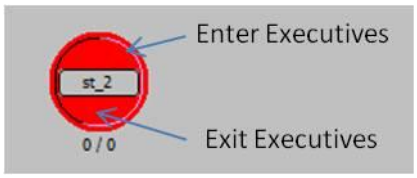


Figure 2.2.4: Two Hot Spots per State.

There are three different types of states: initial State, forced State and unforced State as shown in figure 2.2.5:

- The initial state is the place where execution begins in a process. They are red tokens.
- A forced state does not allow a pause during the process. They are green tokens.
- An unforced state allows a pause during the process. They are red tokens.

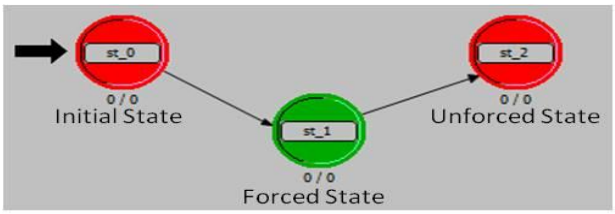


Figure 2.2.5: Different Types of States

- Process Model - Transitions

Transitions describe the possible movement of a process from state to state and the conditions allowing such a change.

- Exactly one condition must evaluate to true
- If the condition statement *(x == y)* is true, the transition executive *(Reset_Timers;)* is invoked

Figure 2.2.6 shows the condition (x==y) associated with a transition between st_2 and st_3.

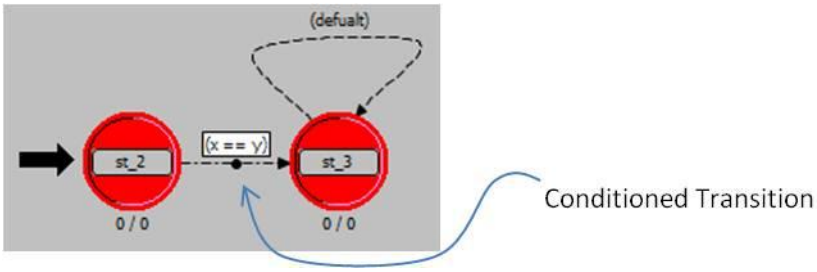


Figure 2.2.6: Transition between st_2 and st_3 with (x==y) condition

- Complete Finite State Machine Application

An example of a wireless node is described in figure 2.2.7.

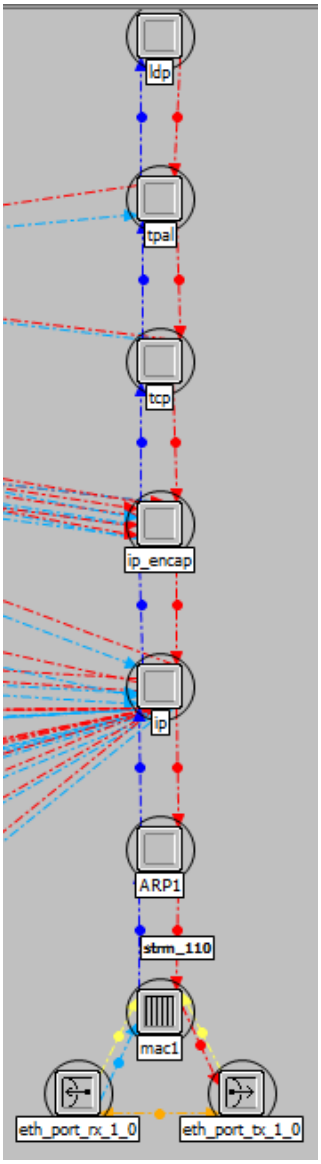


Figure 2.2.7: Wireless Node with 7 Layers (or Process Models)

Process models are represented by finite state machines (FSMs) and are created with icons that represent states and transitions, the logical links between states. The following example reveals the state diagram details of MAC process model (mac1) in figure 2.2.8.

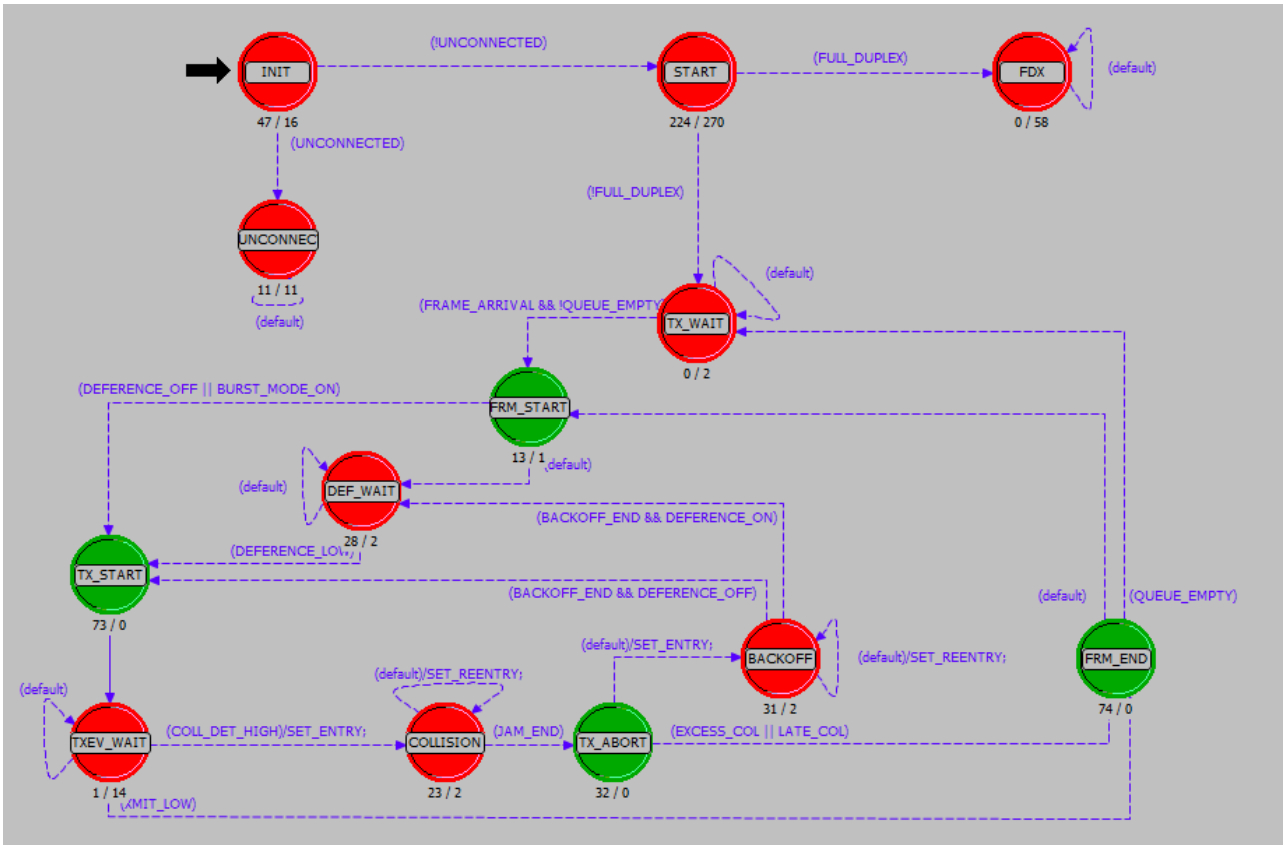
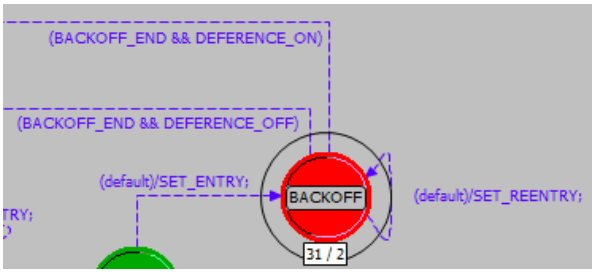
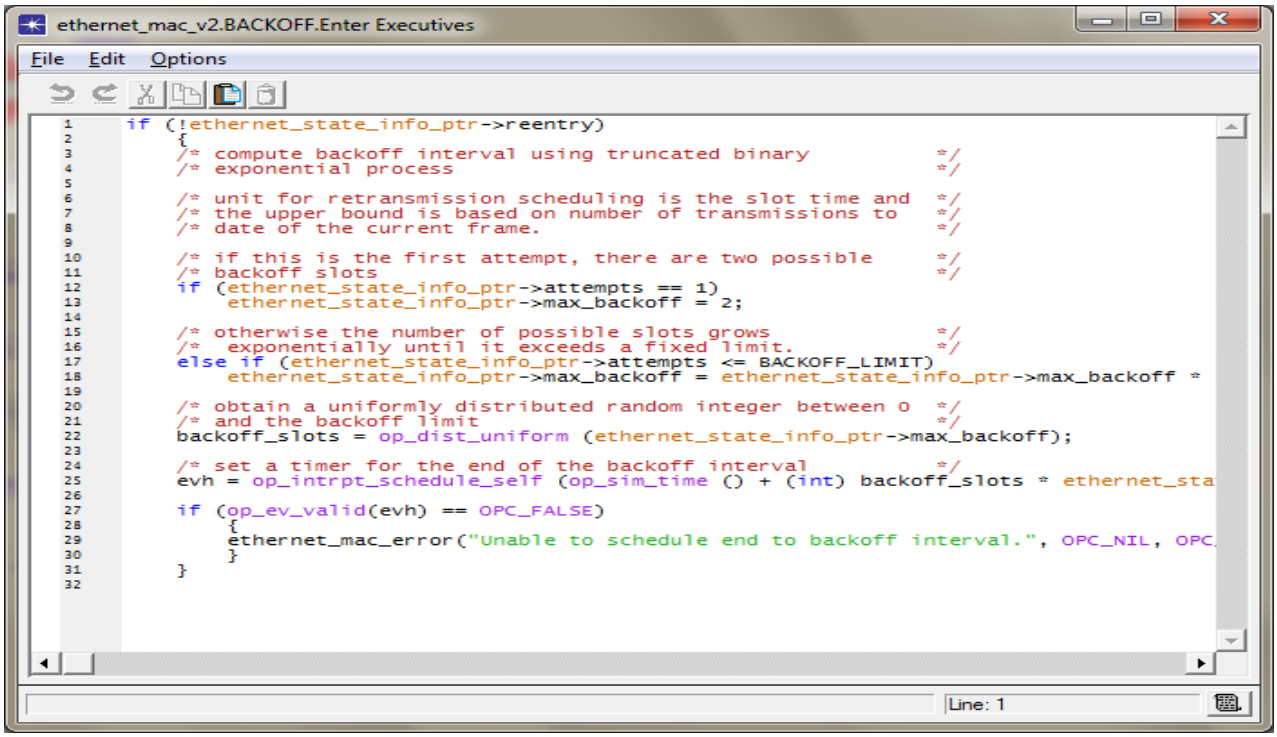


Figure 2.2.8: MAC FSM

Operations performed in each state or for a transition are described in embedded C or C++ code blocks as explained above in this section. Figure 2.2.9(a) shows the Backoff state. Figure 2.2.9(b) present the code associated with Backoff algorithm.



(a)



(b)

Figure 2.2.9: (a) Backoff Process Node, (b) C++ code With Opnet Interrupts for Backoff

Within Opnet environment, there are number of editors that allow the creation of all required details to model and simulate a complete network setup.

There are two possibilities how to interconnect WiMAX Opnet model with real network equipment. The first possibility is ESD system. The second is called System In The Loop (SITL). There are advantages and disadvantages associated with each of them:

- External System Domain:

External System Domain (ESD) is an external system is Opnet's representation of a model whose behaviour is determined to Opnet. Such a model can be anything from a microchip to

a model of user behaviour pattern. The Single-Process Co-simulation is shown in figure 2.2.10.

Figure 2.2.10: Single-Process Co-simulation

The interface between Opnet kernel and the process object is set of interrupts/functions provided by Opnet. There are over 300 of them. To be familiar with each of them, the trial and error is the best approach albeit time consuming.

- System In The Loop (SITL)

SITL is a separately distributed library for Opnet Modular which provides an interface to link real network hardware or software applications to the OPNET discrete event simulation. External devices are connected to the simulation loop over SITL gateways operated as a bridge interface between the simulation environment and the network interface of the host computer. Packets transmitted between the simulated and real networks are converted between real and simulation formats.

The SITL module is mainly focused on real-time communication with devices based on the Ethernet technology while the use of ESD system is much more versatile (Opnet 17.1, 2012). Hence, Weather Station is implemented in this project using ESD system.

2.3 Results of WiMAX Model

The network model shown in figure 2.3.1 is a one cell base station WiMAX with five subscribers. The backbone is the service provider. Some of the subscribers modules are smaller than other. This is signifying the distance from the base station. Further away from BS, smaller the unit appears on worktop of the simulator.

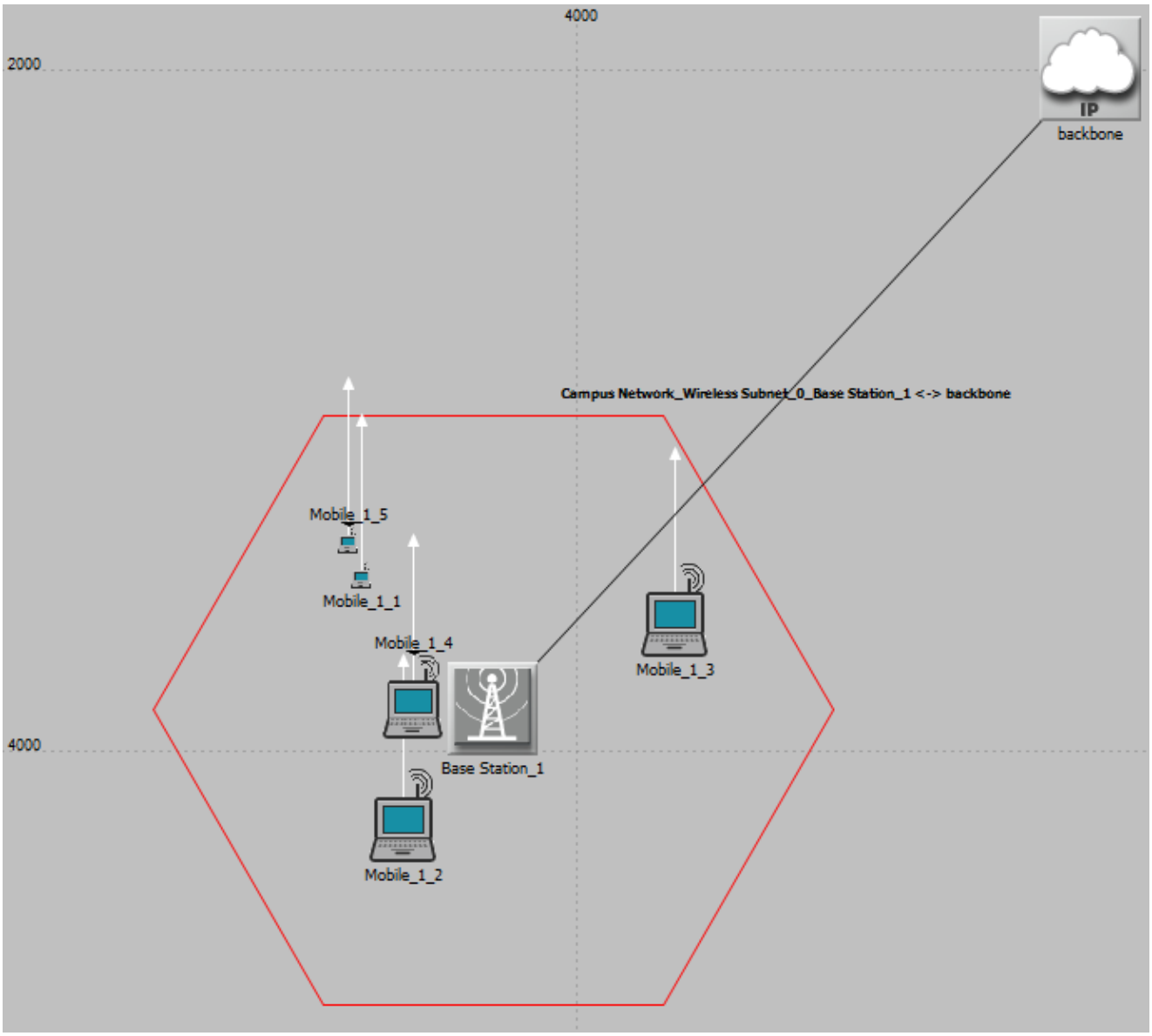


Figure 2.3.1: One Cell Base Station WiMAX

All characteristics of WiMAX listed in section 1.4.15 are included in this model. Within this stage, the two main characteristics of WiMAX have been selected: Delay per second and Throughput per packets/second. The simulation will collect the data for these two items. This is shown in figure 2.3.2.

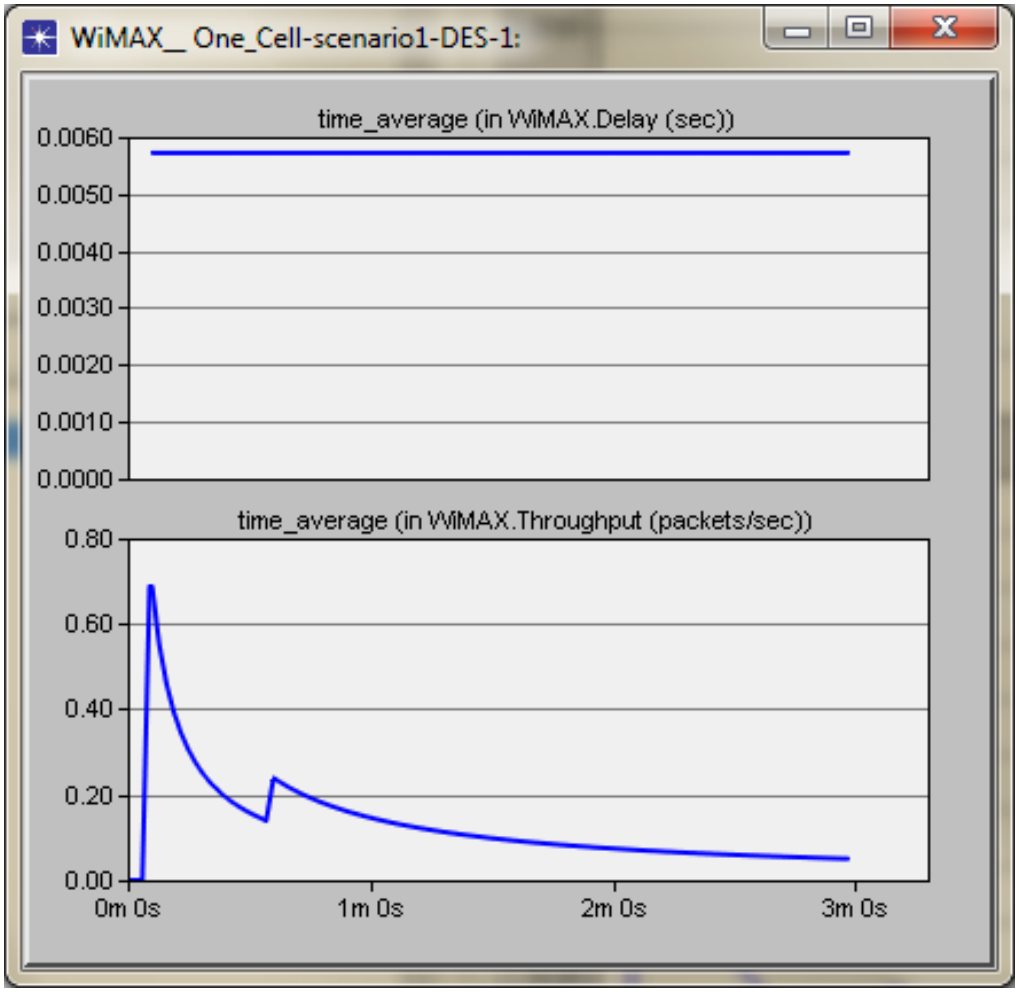


Figure 2.3.2: Opnet Simulation Results for One Cell WiMAX

The depicted results are based on two parameters, the Delay and Throughput. Both have been outlined on Chapter 3, section 3.1.

As the delay is fixed from the IP backbone to Base Station to Subscribers and from Subscribers to Base Station to IP backbone with value 0.006 second, the throughput value ranges from 0 to 0.7 packet per second.

2.4 Chapter Summary

This chapter provides details of Opnet concept. It discusses the structure of the architecture of Opnet and provided the main components that are required to model a network system. Opnet produces results in terms of reports and graphs in single or multiple formats. Next chapter presents the CloneWAN network model including the architecture, simulation and results.

Chapter three: CloneWAN Model Architecture, Simulation and Results

The previous chapter, chapter two provided details at modular level to the software platform that is intended to be used for this research project. This is Opnet version 17.1.

The proposed CloneWAN model architecture, simulation and results are discussed in this chapter. CloneWAN is a complete network system distributed over the UAE. The power and the transmission distance have been considered and accurately simulated. The simulation included one weather station.

Section 3.1 presents the architecture of the model. It shows that the Weather Stations (WS) are connected with the IP cloud directly. Base Stations (BS) do follow WiMAX standard that specified on IEEE 802.16e protocol. The details of the base station have been furnished in section 3.1.2. This model is suggesting that the two units, WS & BS follow the standard TCP protocols. As the warning messages is streamed and coded with the weather stations, servers and mobile users represent the other end of the spectrum of the proposed network. This is presented in section 3.1.3. To complete the details of the CloneWAN model, Opnet analysis have been presented in section 3.1.4. The section ends with a summary in section 3.1.5.

The full details of CloneWAN are presented in Section 3.2 while Section 3.3 outlined the results of Opnet for CloneWAN architecture.

Section 3.4 is dedicated to the sensor hardware. It provides the details of the DSP interfaced with set of sensors. The sensor collects the temperature and other weather information and sends them to WiMAX Base Station.

3.1 CloneWAN Architecture

The three base stations and Weather Station are connected to IP cloud as shown in Figure 3.1.1. In this example of CloneWAN, 10 subscribers are registered to the network setup. The subscribers are represented as server or mobile users.

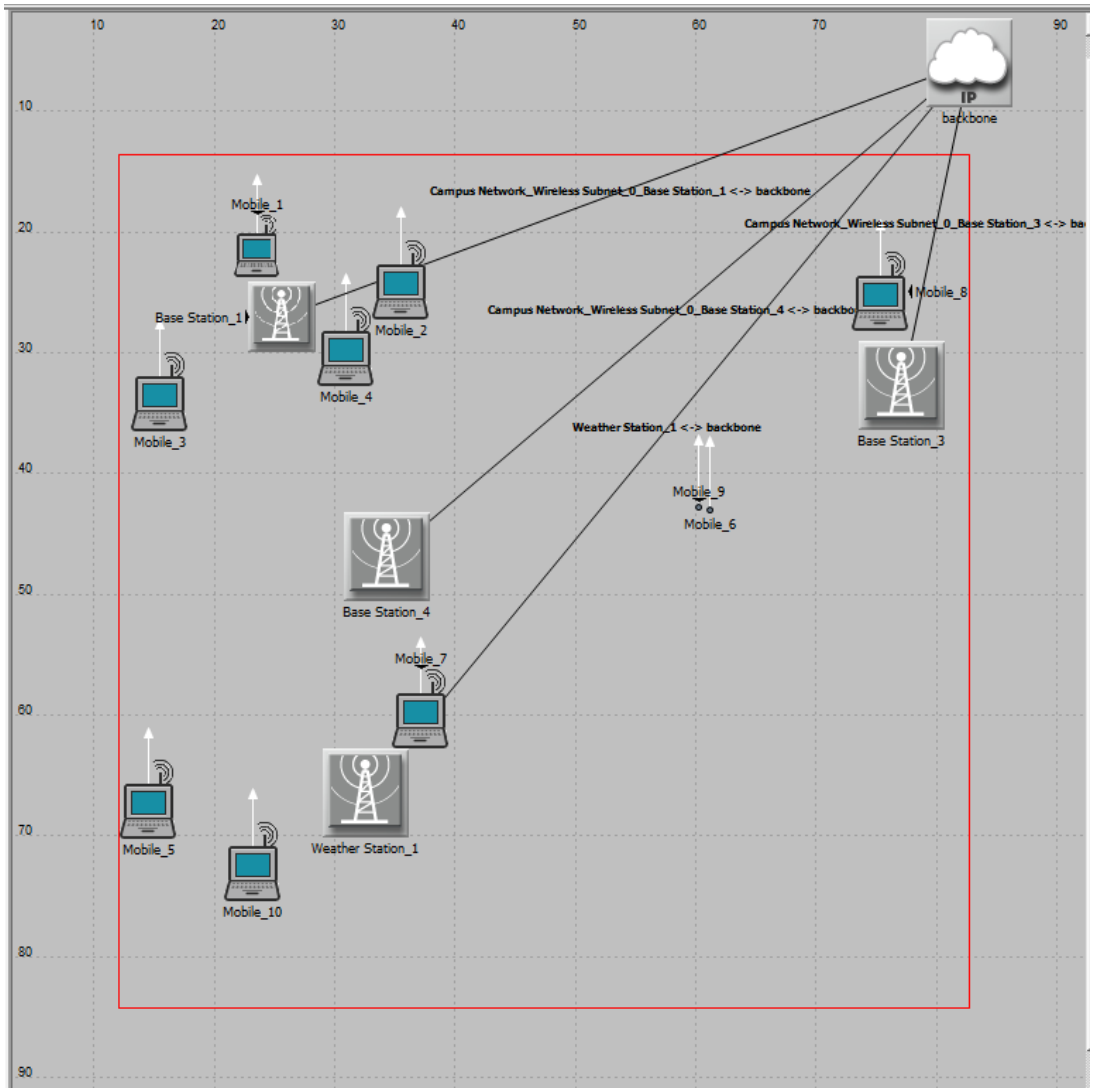


Figure 3.1.1: CloneWAN Example with 3 Base Stations and One Weather Station

3.1.1WiMAX Stations

This section describes the differences between WiMAX base Station and the proposed Weather Station.

3.1.2 Base Station (BS) and Weather Station (WS)

Typical WiMAX base station is shown in figure 3.1.2.1. It consists of wireless base station equiped with a sector antenna and wireless modem (Application Note, 2006).

Figure 3.1.2.1: Base Station: Wireless Base Station Equiped with a Sector Antenna and Wireless Modem (Application Note, 2006)

Weather Station is shown in figure 3.1.2.2. It WiMAX base station with three extra equipment, Clone server, DS-sensor interface and PC-Based Weather Station.

Figure 3.1.2.2: CloneWAN Station: Wireless Base Station Equiped with a Sector Antenna, Wireless Modem, Clone server, DS-sensor interface and PC-Based Weather Station

Base Stations and Weather Station within Opnet CloneWAN model are shown in Figure 3.1.2.3.

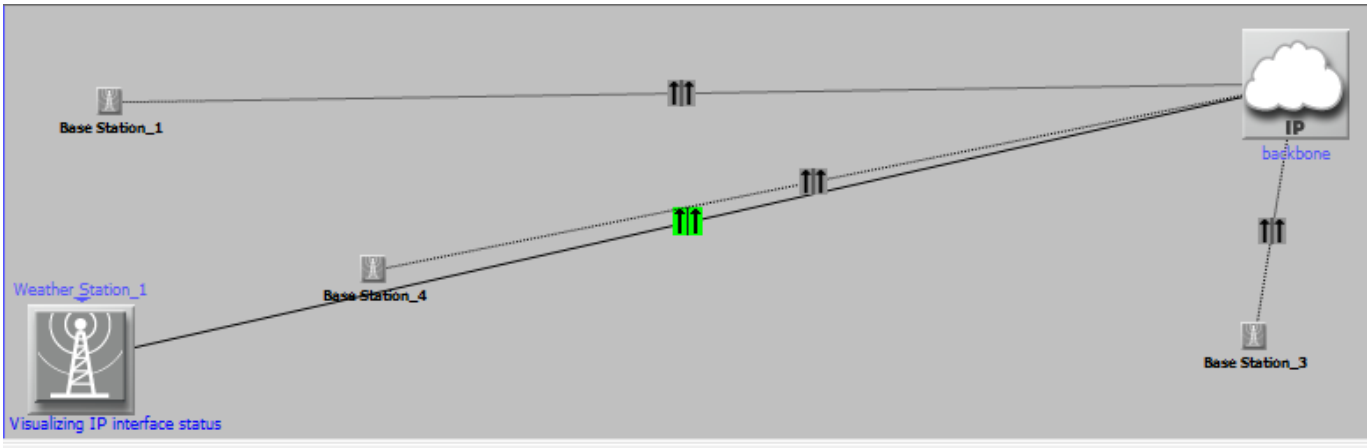


Figure 3.1.2.3: CloneWAN Base Stations and Weather Station

The antenna gain for BS & WS is defined as -15dB. The antenna is modelled to take into consideration the local environment of UAE, i.e. the flatness, the dampness and the heat.

Power Consumption

The maximum number of subscribers can register to one station, including weather station is 100 as the power of each station has been set to 0.5 Watt and 0.005 Watt is allocated for transmitting to one subscriber.

The minimum tolerable received power density at the BS is -110 dBm/subchannel while the maximum is -60 dBm/subchannel. The power density is expressed as the amount of power in a single subchannel. This threshold is used to guide power corrections for the transmitted power of an SS associated with this BS.

More specifically, if the MS transmission is received at a power below this threshold, the BS will issue a ranging-based power correction to the MS's transmitted power, such that the received power at the BS will be above this attribute's setting.

In the current implementation, the power correction issued is computed to bring the received power density (per subchannel) to the middle point between the Minimum Tolerable Power (this attribute) and the Maximum Tolerable Power.

The number of codes for initial ranging CDMA, handover ranging CDMA, periodic ranging CDMA and bandwidth request CDMA are 8 as indicated in Table 353 of the 802.16 Standard (IEEE, 2005).

Ranging Backoff

802.16x (x is any letter between a to y) standard implements Exponential Backoff algorithm (EBA) (Nuaymi, 2007). Timeout (the end to continue sending messages) is an issue with

network technology. EBA offers more chances to send the message up until success of delivery. The algorithm is expressed as:

If *c* represents the number of collision or timeout and *N* the maximum number of Backoff slots, then

$$N = 2^c - 1$$

Hence,

The Backoff time Possibilities = BtP $= \frac{1}{N+1}\sum_{i=0}^{N} i$

For example, if the fourth Timeout (c=4), the number of Backoff slots is:

$N = 2^4 - 1 = 15$

Therefore, the expected Backoff Time Slots is:

$$\text{BtP} = \frac{1}{16}(0+1+...+15) = \frac{119}{16} = 7.4375$$

This is almost equivalent to either the average of minimum and maximum number of slots or half of the maximum number of Backoff slots:

$$\text{BtP} = \frac{2^c - 1}{2} = \frac{N}{2} = \frac{15}{2} = 7.5$$

So, for network application, there is a need to provide two values, the minimum and maximum Range Backoff slots.

The initial range over which the ranging backoff window is picked randomly has been calculated per station. The range doubles with each transmission perceived by the SS to have failed (up to a range of "Ranging Backoff End" slots). Only values between 0 and 15 are allowed and it has to be expressed in power of 2.For instance, if this attribute is set to 3, at the first transmission attempt, the backoff window is chosen randomly from across a range of $2^3 = 8$ slots.

The "Ranging Backoff Start" and "Ranging Backoff End" settings are applied to both initial and periodic ranging (when the latter is done via a contention mechanism, as in OFDMA).

The assigned values to backoff parameter are Rang Backoff Start 2, Rang Backoff End 4, Bandwidth Request Backoff Start 2 and Bandwidth Request Backoff End 4.

The Number of Transmitters Per Weather and Base Stations

Each station is SISO, i.e. the number of transmitter per station is one.

BGP

All base station IP Routing Protocol follows EIBGP (external) and IGRP (internal) protocols.

Security

The security has not been specified here as the network setup is not considered subject of threat.

3.1.3 Subscribers or Servers and Mobile Users

Three mobile subscribers are shown in figure 3.1.3.1 within CloneWAN network.

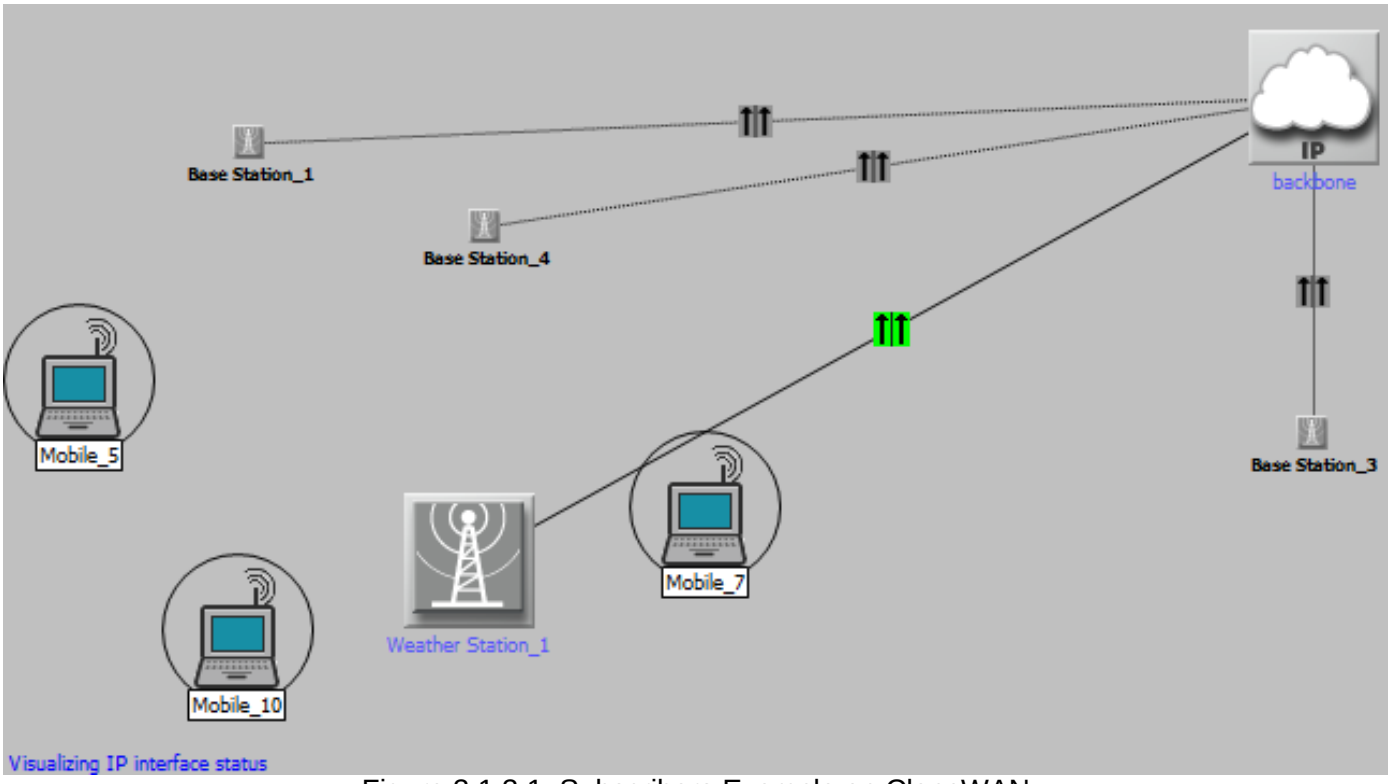


Figure 3.1.3.1: Subscribers Example on CloneWAN

Subscribers Antenna

The antenna gain has been specified to each subscriber as -1dB.

Subscribers OFDM

The physical specifications for base station and the subscribers must be the same. The designated physical specification for the base station is wireless OFDM. Hence, the SS has been assigned Wireless OFDM.

Subscribers DL and UL Buffers and SDU Sizes

The Downlink (DL) buffer is 64Kbyte and the average SDU (Session Data Unit) size is 1500 bytes (Nuaymi, 2007). Similar setup for the Upperlink (UL) has been assigned.

Subscribers Modulation and Coding

According to IEEE 802.16 protocol, the modulation and coding are adaptive.

Terrain Type

The terrain typesetting of the Suburban Fixed pathloss model adjusts the model to the one of the three most common types of terrain found across the United States, as follows:

- Terrain Type A corresponds to hilly terrain with moderate-to-heavy tree densities
- Terrain Type C corresponds to mostly flat terrain with light tree densities
- Terrain Type B corresponds to a compromise between the terrains A and C

The selected terrain option is Type C.

3.1.4 Weather Station Model

The Weather station has been visualised in figure 3.1.4.1.

Figure 3.1.4.1: Weather Station Consists of Hub and Set of Receive and Transmit Nodes

The node level is the top level within Opnet, followed by a lower level, the process level.

The hub node model consists of a point-to-point one receiver to multi transmitter for each peripheral WS node, and a process model used to relay packets from a receiver to the appropriate transmitters.

3.1.4.1 Functions of the Weather Station

The hub node model, shown in figure 3.1.4.1.1, consists of a point-to-point one receiver to multi transmitter for each peripheral WS node. Three different nodes are deployed in the WeS model, the first one is aimed for normal communication, the second one is meant for warning and the third node is for alerts, i.e. it takes higher priority.

The purpose of the weather station model is to simulate packets traveling from IP cloud to all subscribers through the packet switching hub node.

In the hub node it can be assumed that packets containing destination addresses are arrive randomly on the incoming link from the IP cloud. The destination address is an integer value specifying a destination peripheral node. The hub node must contain a process model that can retrieve the incoming packets, read the destination address, and send the packets to the appropriate point-to-point transmitter.

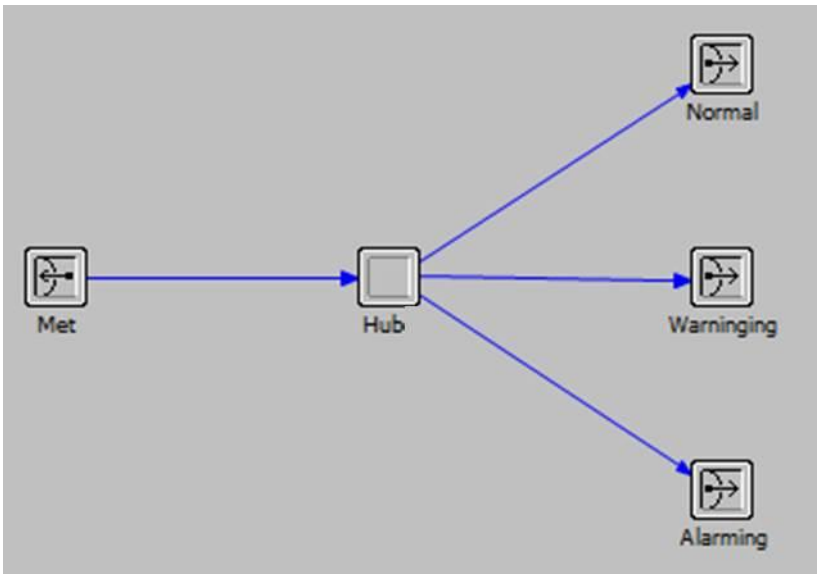


Figure 3.1.4.1.1: The Hub Node as Modelled within Opnet

Figure 3.1.4.1.2 represents the details of the hub node. It is a process model used to relay packets from a receiver to the appropriate transmitters.

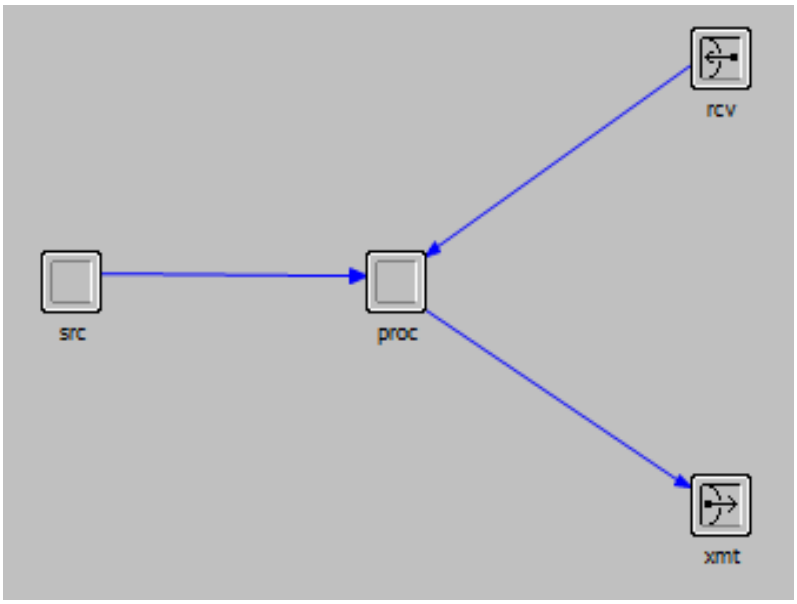


Figure 3.1.4.1.2: The Peripheral Node as Modelled within Opnet

WeS is dedicated to analyse the information received from the Weather Met Office and the weather sensor. For example, if the expected temperature next week is -10 ^{o}C, WS will send a warning message to subscribers with a specific recommendations reference possible damages or danger. Another important example is to alarm subscriber of any natural disasters. However, all these warning messages follow CAP standard protocol (addressed in section 1.5).

Natural disaster predications and weather information have been categorized to three levels:

- Normal
- Warning
- Alarming

Next stage of the research will address this aim and model the above scenarios and processes them within a special unit that is called by Opnet as External System Domain (ESD). ESD is an external system for Opnet to model the behaviour of a microprocessor/microcontroller or DSP. This interface is considered for the proposed Weather sensor.

3.1.4.2 Building WS Network Model

WS network model has been built following the steps listed in figure 3.1.4.2.1, flow chart:

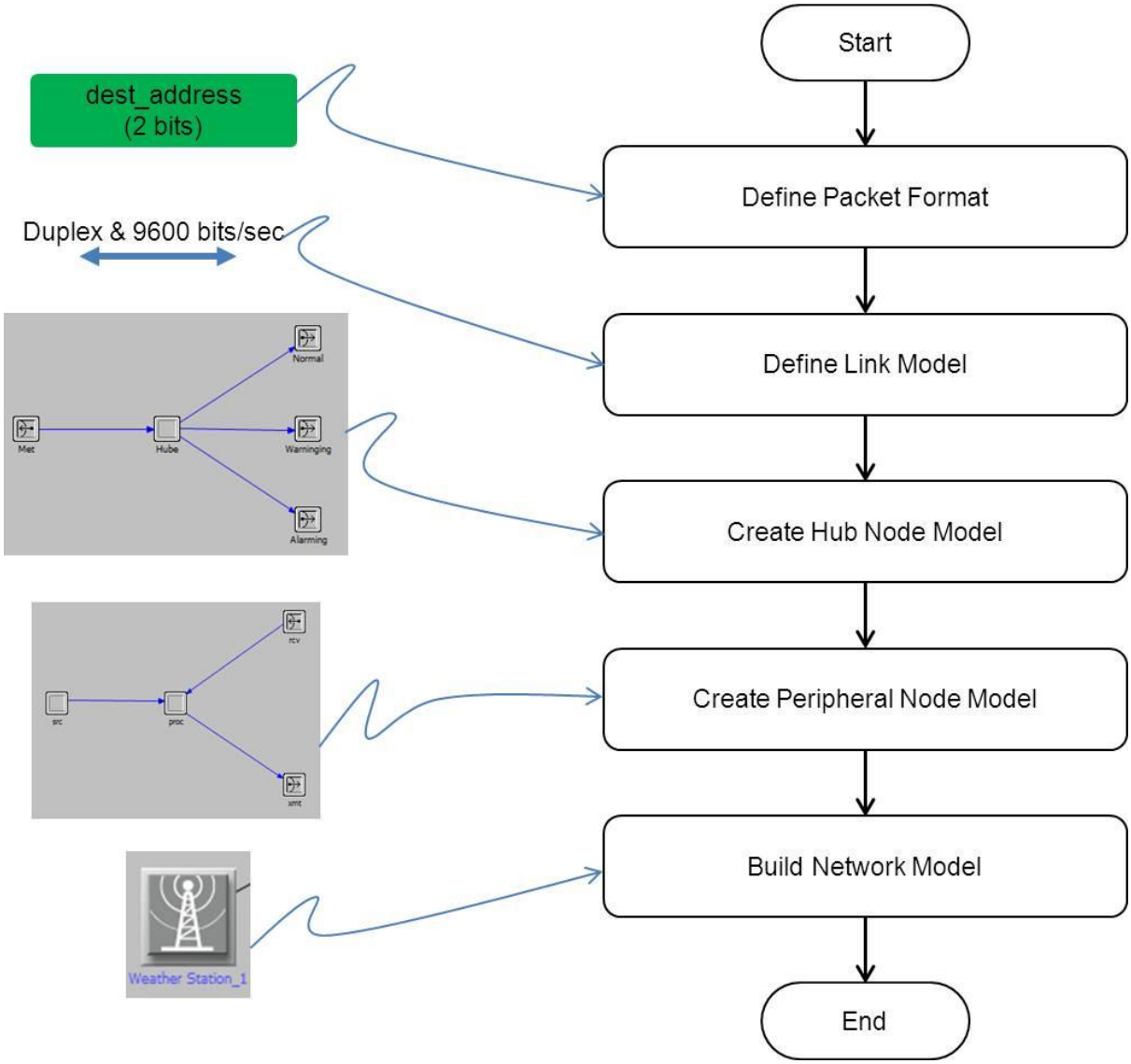


Figure 3.1.4.2.1: WS Model Flow Chart

There are several design concepts that has been considered to build the network model:

- Network topology and the physical communication medium
- The functions of the different node types
- The method the process model uses to determine which point-to-point transmitter addresses a particular peripheral node, and
- The role of peripheral nodes

Each node within Opnet, there is at least one process or at least one finite state machine with associated transitions (conditions). These states specify Opnet macros and C++/C functions. For example, figure 3.1.4.2.3 shows a process model called PK_ARRVL. Here, two main descriptions to be entered separately, the header and the function code.

The header for this unit is described as follows:

```
#define PK_ARRVL (op_intrpt_type () == OPC_INTRPT_STRM)
```

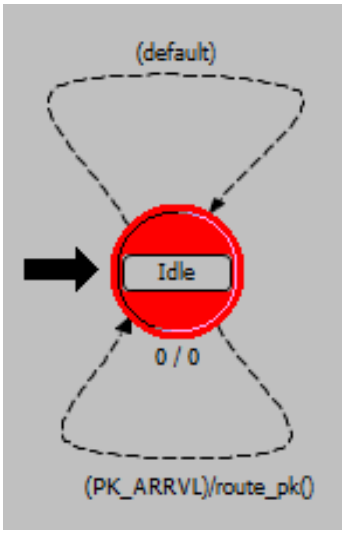


Figure 3.1.4.2.2: Simple Proces Model called PK_ARRVL

The PK_ARRVL condition compares the delivered interrupt type with the predefined symbolic constant OPC_INTRPT_STRM, which indicates a stream interrupt. This is the only expected type of interrupt for this model. To safeguard against run-time missing transition errors, a default transition loop on the idle state, as shown on the above figure.

The function code is listed below:

```
static void route_pk(void)
            {
            int dest_address;
            Packet * pkptr;
            FIN(route_pk());
            pkptr = op_pk_get(op_intrpt_strm ());
            op_pk_nfd_get_int32 (pkptr, "dest_address",
                    &dest_address);
            op_pk_send (pkptr, dest_address);
            FOUT;
}
```

This code executes when the FSM follows the transition. The first line after FIN(route_pk()) has two effects: first, it retrieves the arriving packet from the appropriate input stream (whose index is determined by op_intrpt_strm()). Then, op_pk_get() uses the packet-stream index argument to return a pointer to the packet.

Next, the process must obtain the destination address contained in the packet. The destination address is held in the packet's dest_address field, which is an integer data type. The second line of code assigns the destination address to the dest_address local variable.

This is an experimental stage to the model but the author will develop the complete model within the research project.

3.1.5 Section Summary

This section has presented the full definitions for the WiMAX base station; it highlighted the required parameter for the network base station. This is important as the weather station topology follows IEEE802.16e standard. Hence, it is based on components similar to WiMAX with the addition of Weather Station and Weather Sensor.

3.2 Definitions

The model that simulated on Opnet platform in section 3.1 has produced results in reports and graphs formats. These results are presented in this section.

3.2.1 Total Delay and Throughput Parameters

WAN uses two known applications, Circuit Switching and Packet Switching:

- Circuit Switching

The routers have to be fixed in this type of switching and the path to be dedicated between two stations. Each link, a logical channel is dedicated to the connection to allow the data to be transmitted along the channel as fast as possible. Figure 3.2.1.1 shows Circuit Switching example.

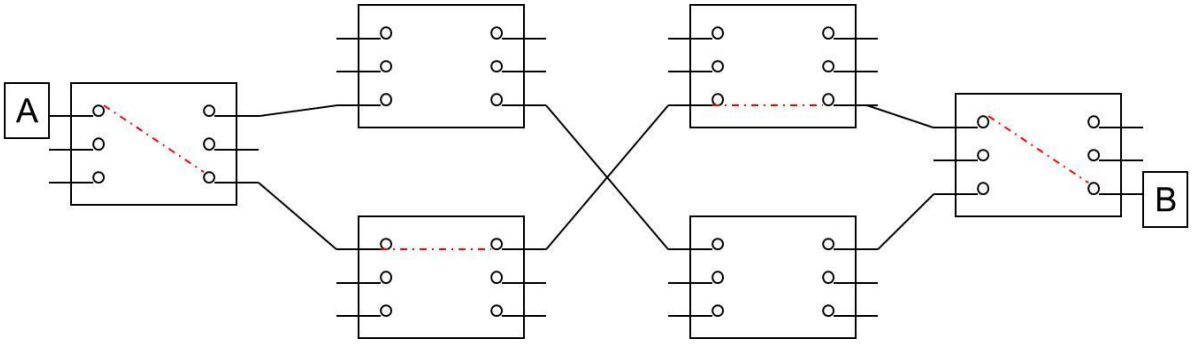


Figure 3.2.1.1: Circuit Switching Example: The red dotted line shows data is traveling from A to B through fixed routers

- Packet Switching

With Packet Switching, data sent in a sequence of packets where each packet passed from node to node along some path but at each node, the entire packet is received, stored briefly and transmitted to next node. This is shown on figure 3.2.1.2. Each packet carries data and headers, such as the Ethernet, IP, and UDP (User Datagram Protocol) as shown in figure 3.2.1.3. If the packet is lost, with another trial the sender will resend. This is repeated up until success with defined number of attempts.

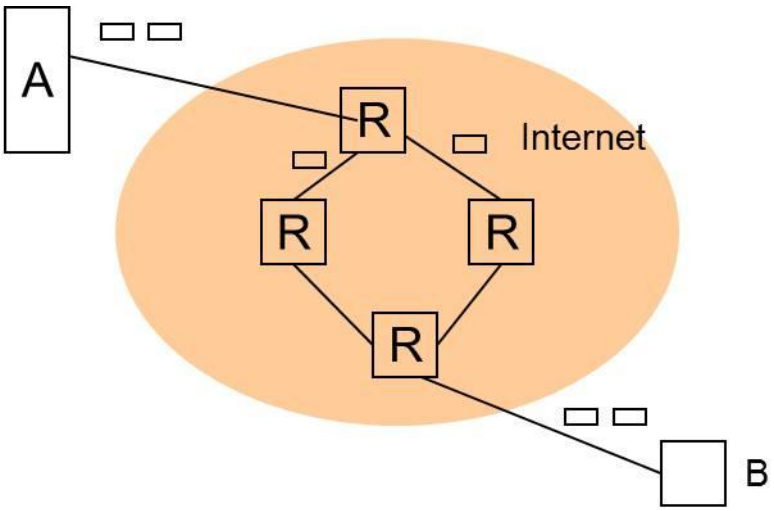


Figure 3.2.1.2: Packet Switching Example

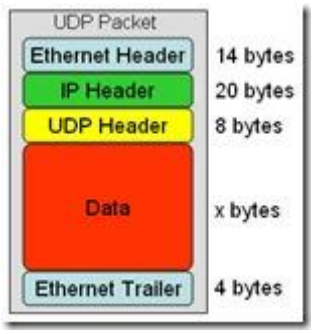


Figure 3.2.1.3: Packet Switching Header

Opnet offers framework to model this behaviour and analyze delays via two parameters, Total Delay and Throughput:

Figure 3.2.1.4 shows the timing events in both cases the circuit and Packet Switching.

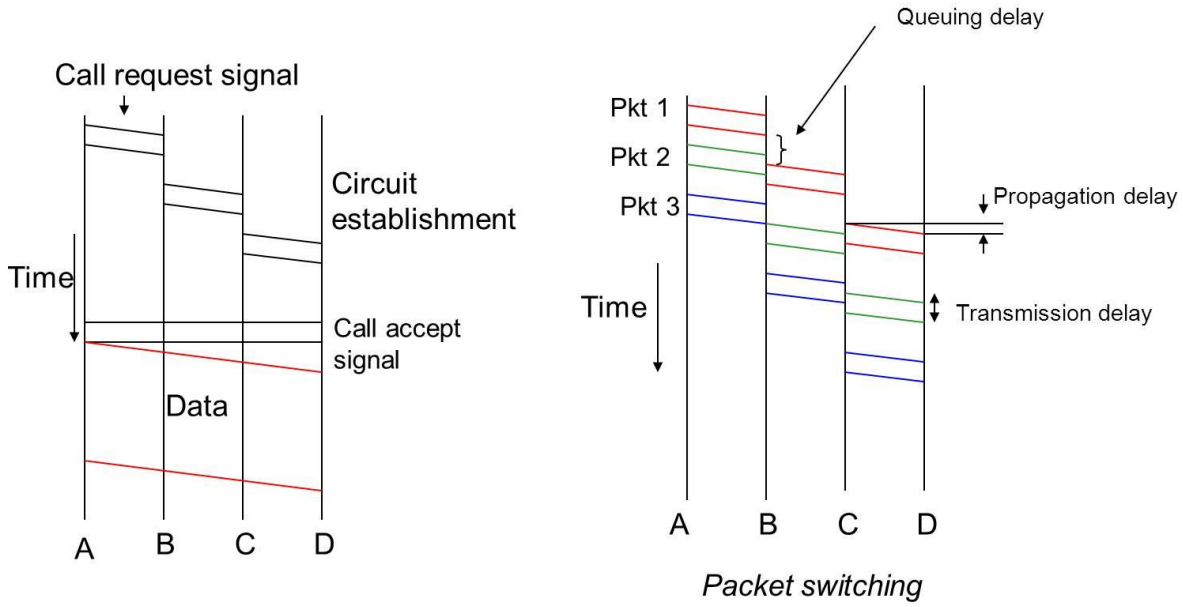


Figure 3.2.1.4: Circuit and Packet Switching Timing Events

With Circuit Switching, after connection is established, information is transmitted at a fixed data rate. Node delays are negligible. Guarantee there will be no losses. On the other hand, Circuit Switching is expensive. It is known for excellent reliability but less efficiency.

Relative to Circuit Switching, Packet switching delays are not negligible. Figure 3.2.1.5 shows Packet switching delays.

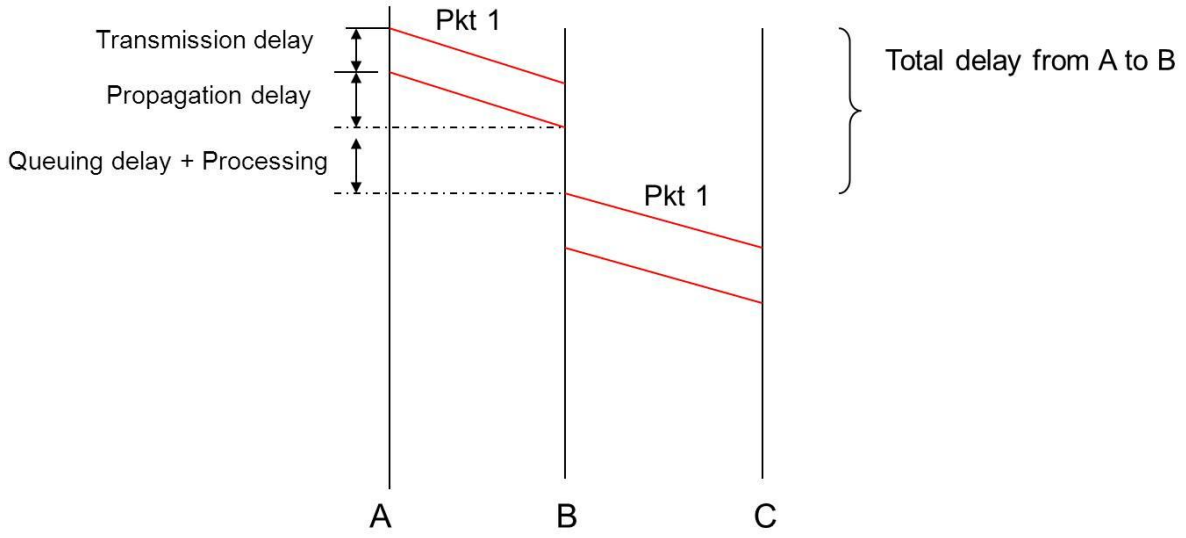


Figure 3.2.1.5: Packet switching delays

Under heavy use there can be a considerable delay with Packet Switching. As data packets can get lost or become corrupted, protocols are needed for a reliable transfer. It is known for excellent efficiency but less reliable.

If L = size of packet (bits) and T is the in seconds, then:

Throughput = L / T Packet per second

That means, as delay increases, the throughput decreases. Other issues impacts the throughput are Packet losses and the type of recovery mechanism.

However, Stallings (Stallings, 2004) specified that:

Smaller packets => higher service rate (or smaller transmission times)

This is based on the following formula titled:

If Service rate on the outgoing link = μ then

$$\mu = \frac{Capacity(bits/\sec)}{Packet\_size(bits/packets)}$$

Virtual Circuit Packet Switching

With Datagram Packet Switching (called connectionless switching), base station requires no logical connection, i.e. no dedicated route for the conversation or session.

As the demand for speech and data communication increased, a newer version of switching emerged to provide both circuit and packet methods in one router, called Virtual Circuit Packet Switching. The Internal Virtual Circuit Packet Switching is shown in figure 3.2.1.6.

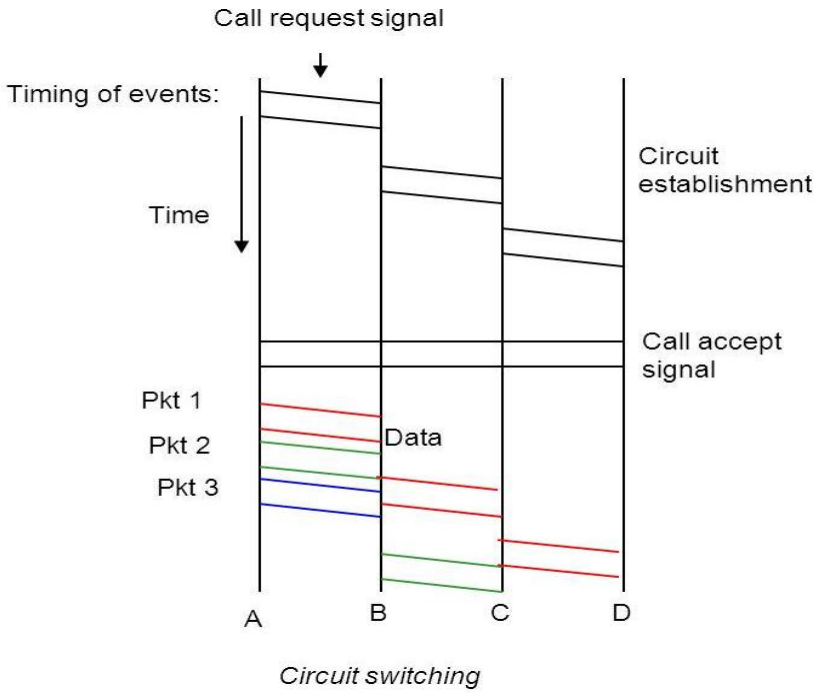


Figure 3.2.1.6: Internal Virtual Circuit Packet Switching

Virtual Circuit Packet Switching sometime called connection switching. The base station requires logical connection, i.e. requires dedicated route for the conversation or session. However, this adds extra call setup delay.

External Virtual Packet Switching, shown in figure 3.2.1.7, is used in PtP connections, i.e. base station to base station connections. It is a reliable data transfer cross the physical link.

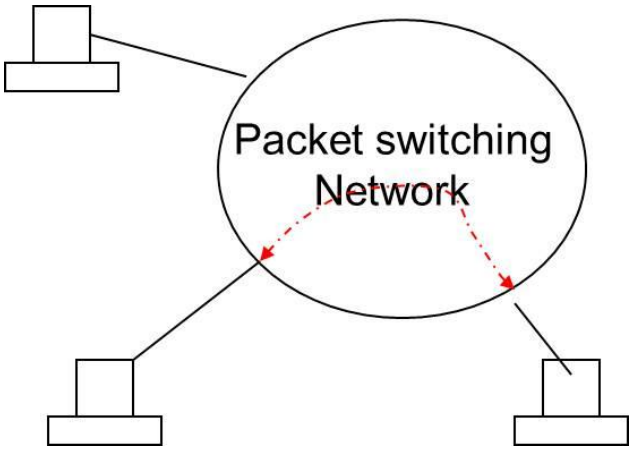


Figure 3.2.1.7: External Virtual Circuit Packet Switching

The Delay and Throughput parameters for a specific traffic have been demonstrated in the result section, section 3.3.

3.2.2 Traffic

Opnet allows creating traffic to be loaded to the network simulation environment. The steps to develop the traffic on CloneWAN are:

- Create application
- Create profile
- Workstation object - Assign profile to desired workstations – In this case the Weather Station and one Base Station have been assigned with specific profiles
- Server object – assign service type (application)

CloneWAN model over UAE is shown in figure 3.2.2.1.

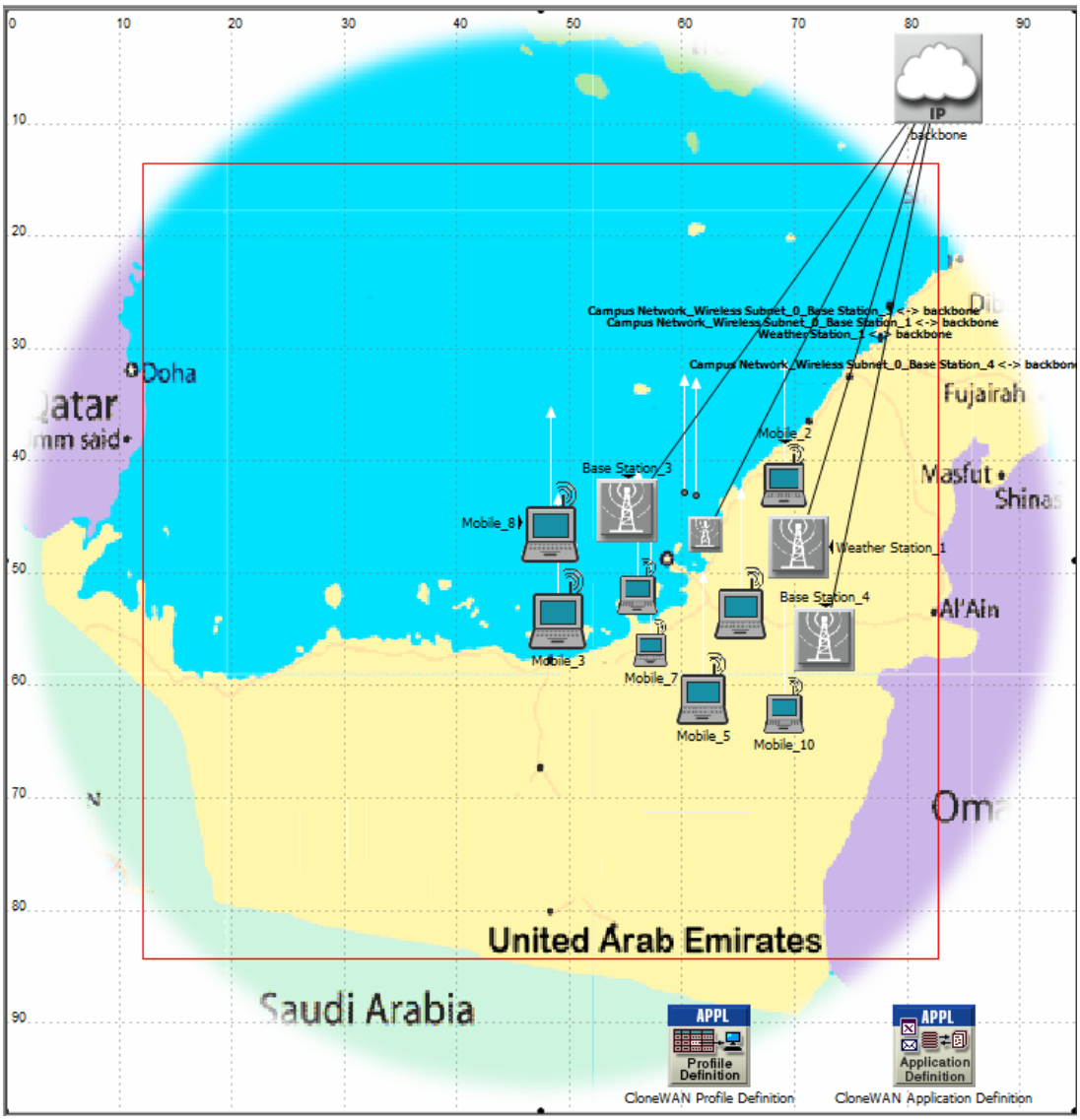


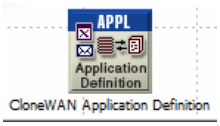
Figure 3.2.2.1: CloneWAN to be deployed over UAE

The network consists of 3 BSs and one WeS, 10 SSs, and 4 backbones link with up to 35Mbps. The area selected for CloneWAN is in UAE over the capital, Abu Dhabi.

Application Definition, Profile Definition, all backbone links, Weather Station, one Base Station and one Subscriber Station are defined here:

3.2.2.1 Create application - Application Definition

The simulation environment provides simple traffic sources, or complex protocols, or discrete set of tasks through set of applications. The examples of applications are:



- FTP
- Email
- Remote Login
- Video Conferencing
- Database
- HTTP
- Print
- Voice
- *Custom*

The "Application Definition" attributes window, shown in figure 3.2.2.1.1, can be used to define the application specifications, MOS and specify voice scheme from one to all.

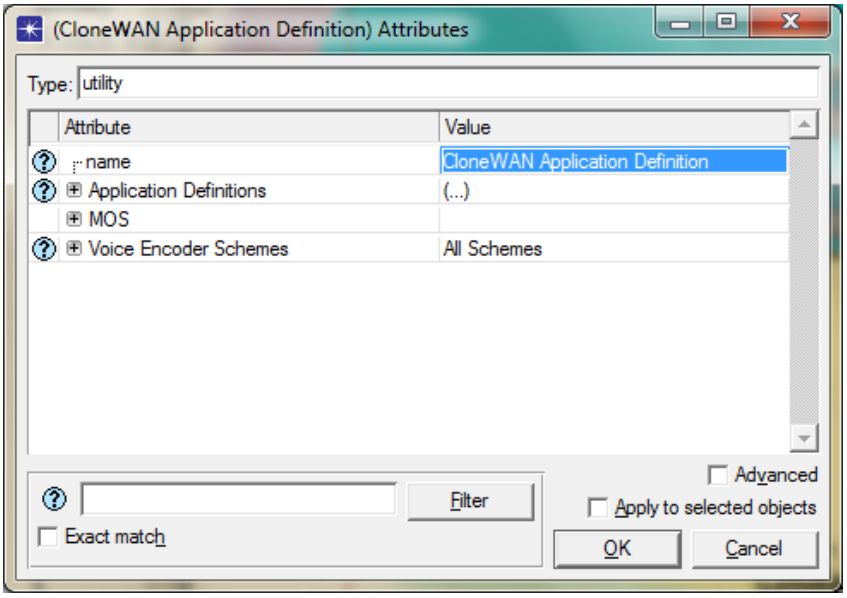


Figure 3.2.2.1.1: The Attributes to CloneWAN Application Definition are Application Definitions, MOS and Voice Encoder Schemes

The three attributes to Application Definition are a) Application Definitions, b) MOS and c) Voice Encoder Schemes:

a) "Application Definitions":

It specifies applications using available application types. It specifies a name and the corresponding description in the process of creating new applications.

For example, "Web Browsing (Heavy HTTP 1.1)" indicates a web application performing heavy browsing using HTTP 1.1.

The specified application name will be used while creating user profiles on the "Profile Config" object.

For CloneWAN, we set the attributes to three applications: HTTP (day time users), Database (night time users) & Voice (weekend users). This is shown in figure 3.2.2.1.2.

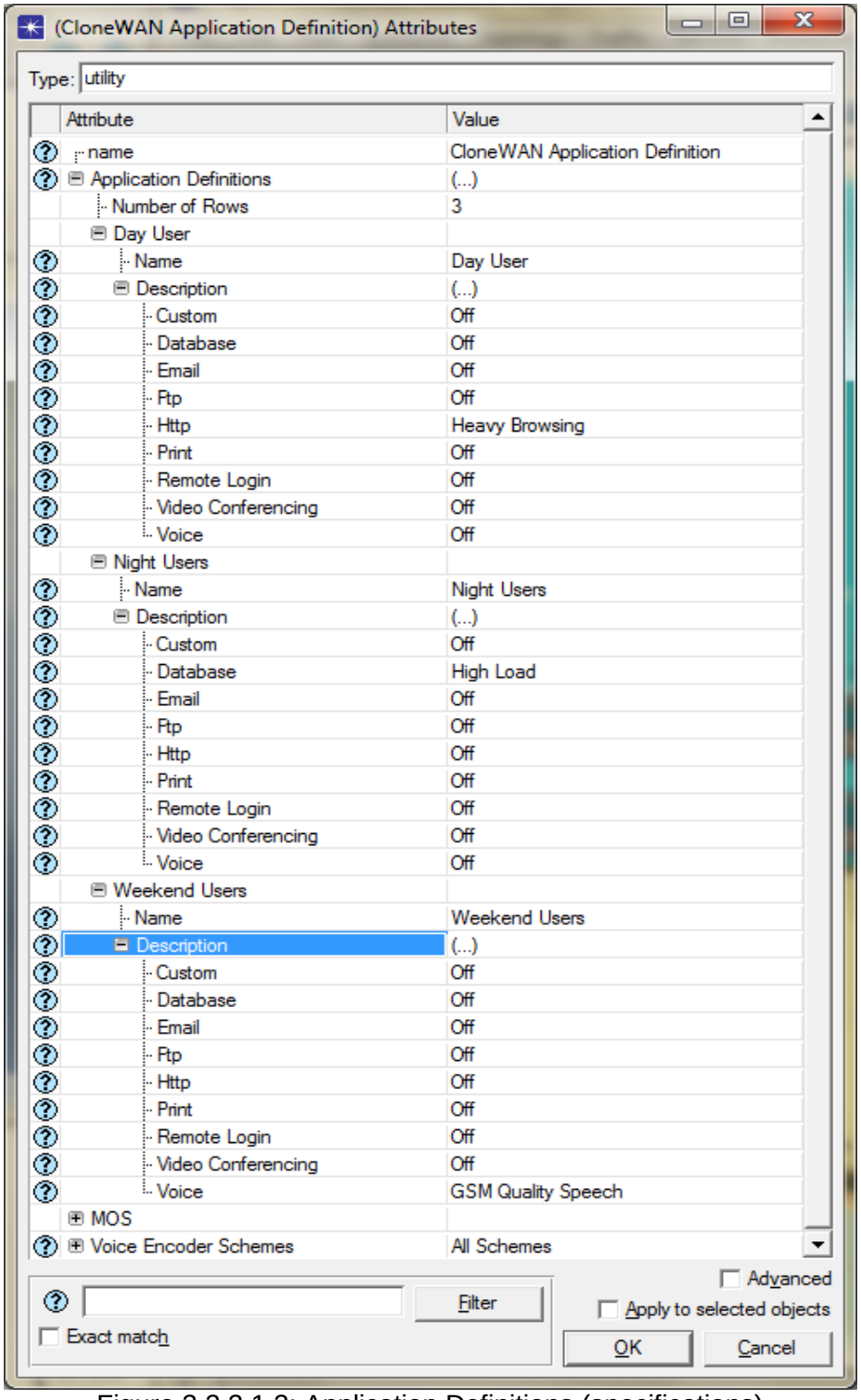


Figure 3.2.2.1.2: Application Definitions (specifications)

b) MOS

MOS (Mean Opinion Score) is a qualitative measure of voice quality on a telephone network. Figure 3.2.2.1.3 shows the attribute the MOS attribute to CloneWAN network.

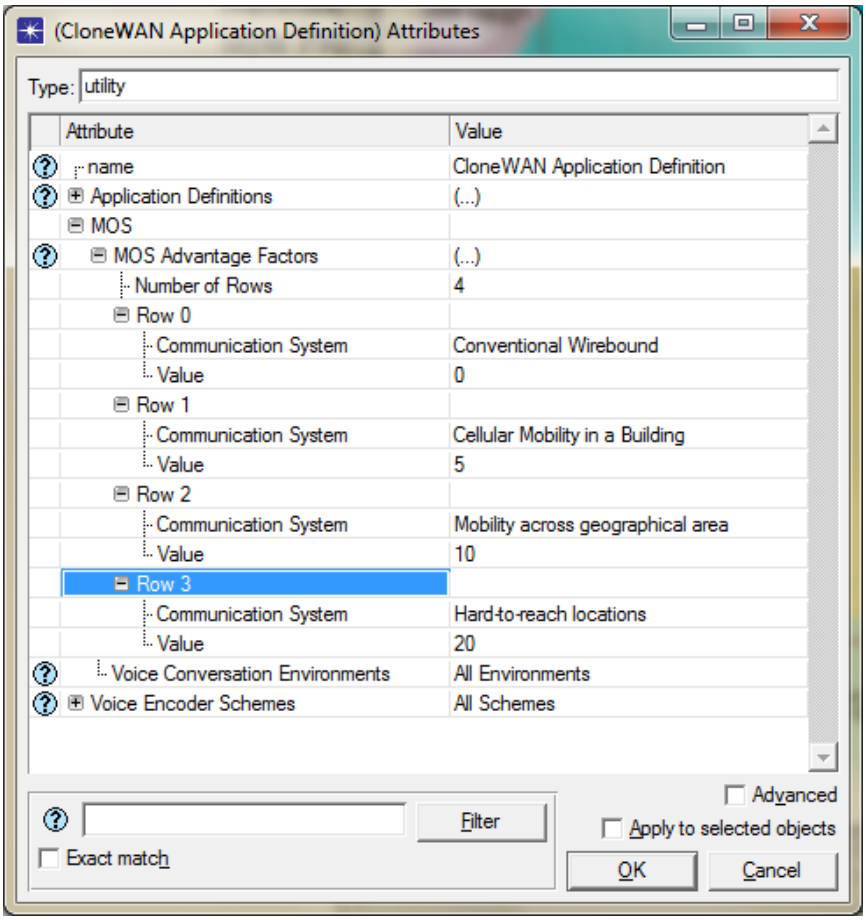


Figure 3.2.2.1.3: MOS Settings

The setting for MOS for different types of Voice Communication Media, Conventional Wirebound, Cellular (Mobile) Mobility in a building, mobility across geographical area and Hard to reach locations. The given associated values are 0, 5, 10 & 20.

c) "Voice Encoder Schemes":

WiMAX protocol implements four voice schemes, G.711, G.729 A, G.723.1 and GSM. If the user is making use of his/her own scheme, the network allows that as well.

Figure 3.2.2.1.4 shows attributes to Voice schemes.

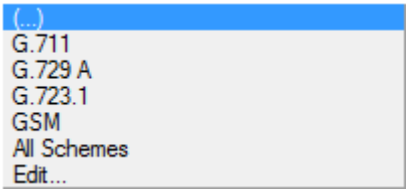


Figure 3.2.2.1.4: Voice Schemes

CloneWAN has chosen PCM – G.711 scheme as shown in figure 3.2.2.1.5 among the rest of the list of speech encoding schemes.

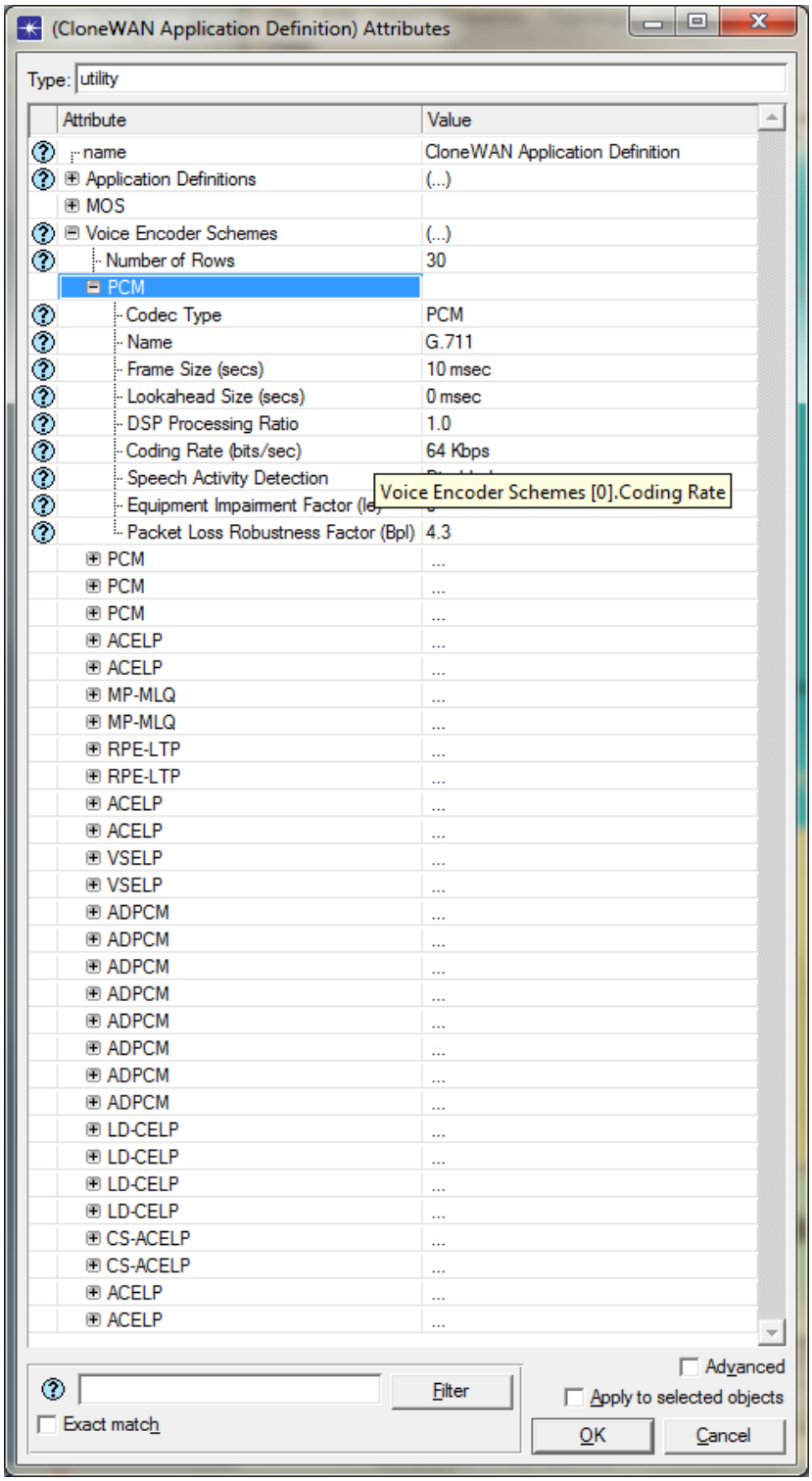
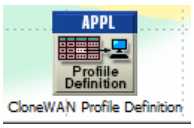


Figure 3.2.2.1.5: Speech Encoders Schemes

3.2.2.2 Create Profile - Profile Definition



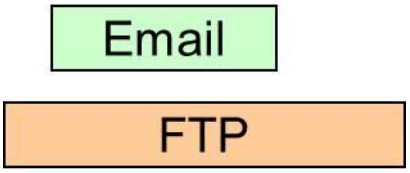
Each customer of WiMAX may follow specific requirements. These requirements have been set in profile icon as defined by Opnet. The aim of the profile is to:

- Set the application that the user uses
- The duration and the cycle of repetition
- Set start time, duration, repeatability, parallel applications or serial applications
  For example:

Serial

or

Parallel



- Workstation object - Assign profile to desired workstations – In this the Weather Station and one Base Station have been assigned with the following profile:
- specific profiles
- Server object – assign service type (application)

Profile Definition Attributes window for CloneWAN is shown in figure 3.2.2.2.1.

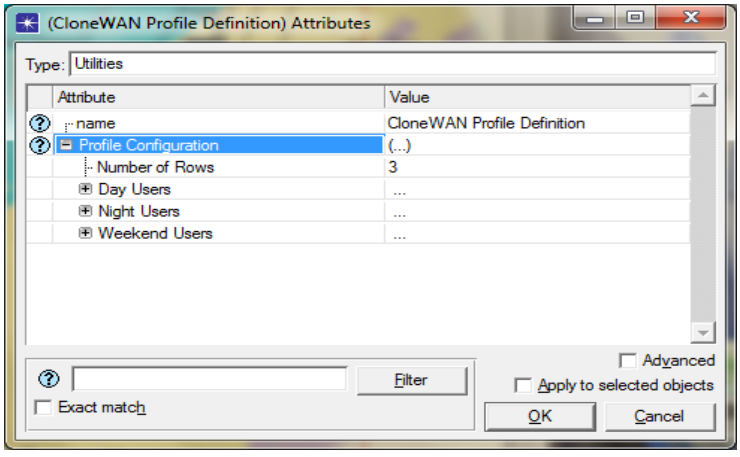


Figure 3.2.2.2.1: The Day Time Users attributes are shown

CloneWAN Profile has been configured to three users; they are labelled as Day Users, Night Users & Weekend Users. The Day Users profile is listed in figure 3.2.2.2.2.

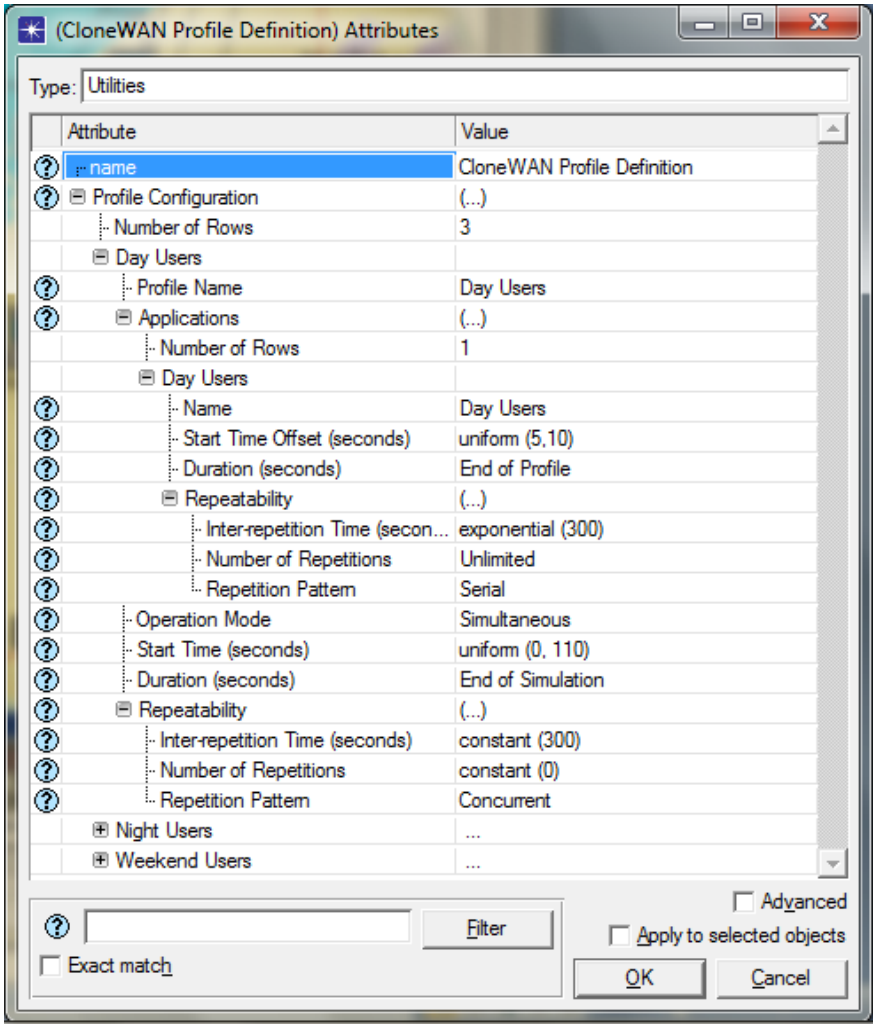


Figure 3.2.2.2.2: The Day Time Users attributes are shown

Initially we assigned one application to Day users. However, the Profile Definition allows any reasonable number of applications per user.

Repeatability specifies the parameters used to repeat execution of this profile.

The Operation Mode defines how applications will start:

- Serial (Ordered) - They can start one after each other in a ordered manner (first row to last row).
- Serial (Random)- They can start one after each other in a random manner.
- Simultaneous- They can start all at the same time.

The start time for Day users is 0 sec and will stop by the end of the profile. This means that the maximum amount of time allowed for an application session before it aborts. This is often

used as a timeout. Here it is selected as End of Profile to end when the profile duration has expired.

Other users carry similar profile.

3.2.2.3 Backbone Links

CloneWAN uses 4 links as shown in figure 3.2.2.3.1.

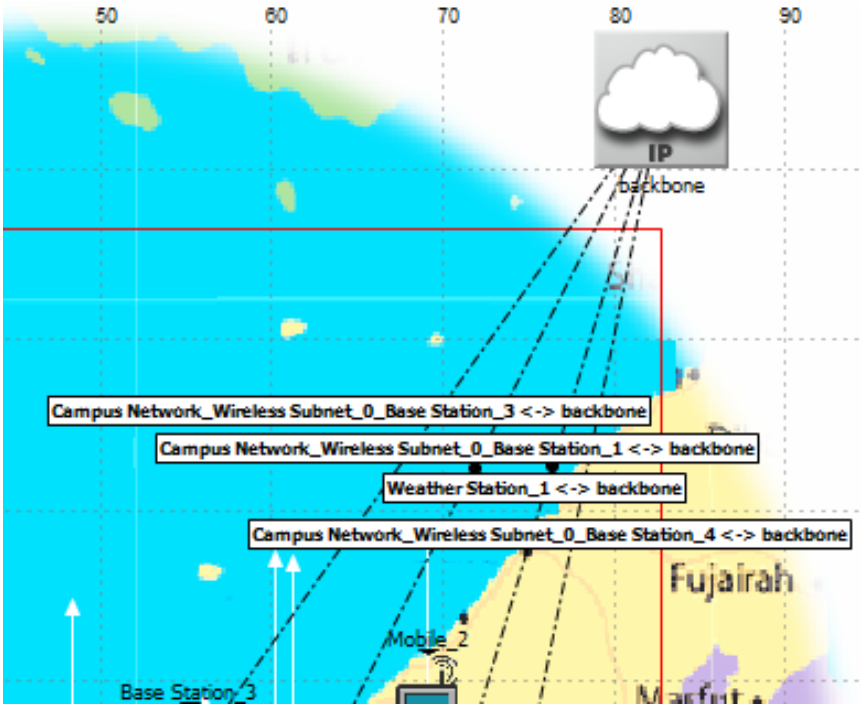


Figure 3.2.2.3.1: All Backbone Links are selected

These network links are duplex point-to point links. Delay value has been assigned to each link. However the speed has been decided as the light speed with each of these links.

The window shown in figure 3.2.2.3.2 is for one of the backbone links. It shows the configuration of the links one by one.

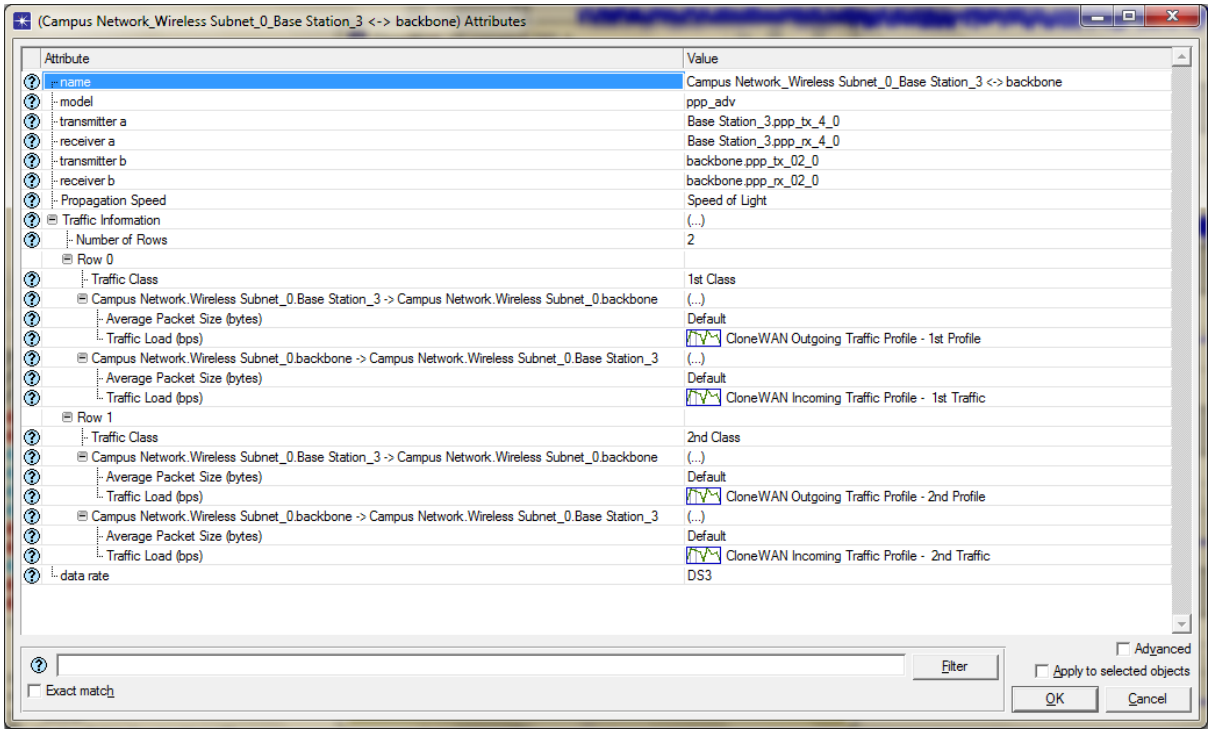


Figure 3.2.2.3.2: Backbone Link Window Attributes

Each attribute within backbone link window is explained below.

Traffic classes are used for classifying packets for QoS handling. Traffic classes referenced here are defined on routers/workstations under the "IP QoS Parameters -> Traffic Classes" attribute.

The average size of the packets making up the background load in one direction of the link is 576 bytes. Other choices are ATM, Frame Relay and Ethernet.

Propagation Speed attribute specifies the propagation speed (in meters/sec) for the medium. It is set to Speed Light. The exact speed of light in vacuum is 299,792,458 meters/sec.

Traffic Information attribute specifies mean baseline loads for this link. A delay representing the effects of the background load is calculated for each explicitly modelled packet transmitted on this link.

Two Traffic Classes have been specified to each link, 1st Class and 2nd Class. The average of the packet size is 576 bytes. The CloneWAN Outgoing Traffic Profile - 1st Profile, i.e. Traffic Load in bps is shown in figure 3.2.2.3.3. Just as a reminder ATM packet size is 1500 bytes or 12000 bits.

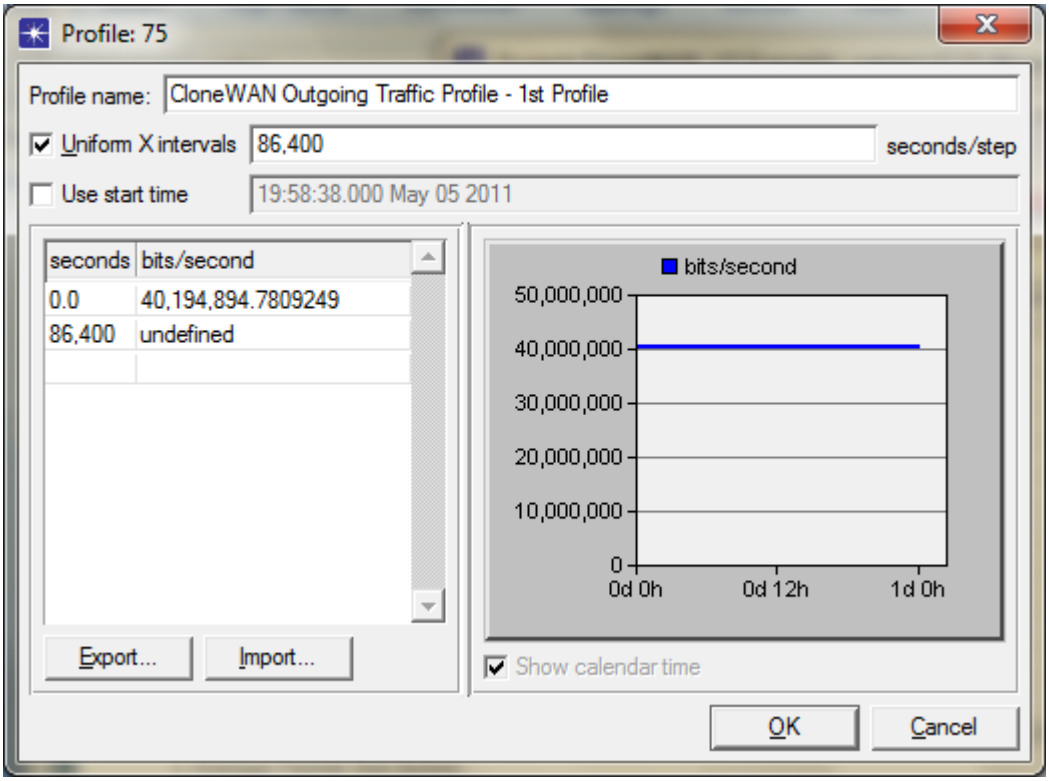


Figure 3.2.2.3.3: Traffic Load 1 (Traffic Profile)

The intervals are uniform and specified as 86,400 seconds per step. Initial rate has been assigned as 40,194,894.7809249 bits per second (just above 40Mbps). The same profile is repeated for the incoming load.

The CloneWAN Outgoing Traffic Profile – 2nd Profile is shown in figure 3.2.2.3.4.

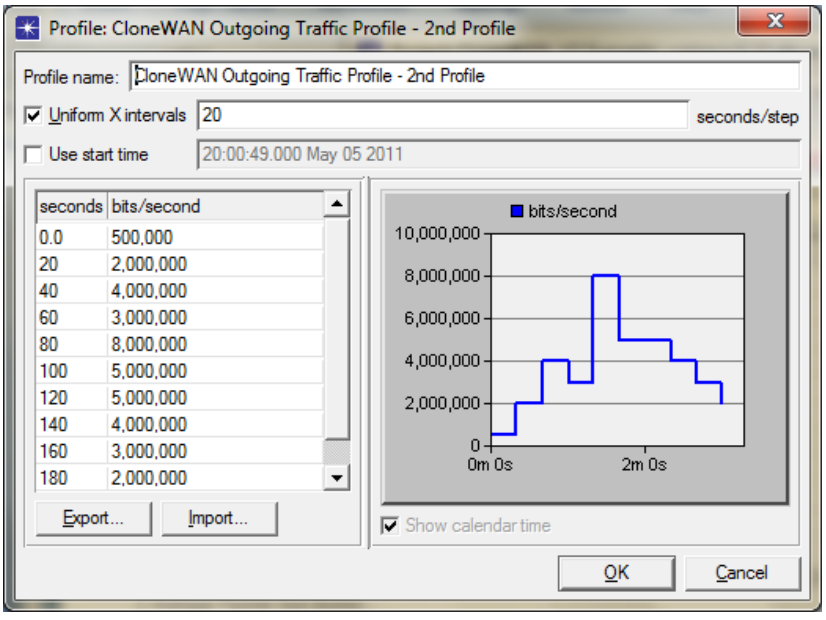


Figure 3.2.2.3.4: CloneWAN Outgoing Traffic Profile - 2nd Profile

For this traffic profile, the interval is 20 seconds per step but runs up to three minutes (3 * 60 = 180 seconds).

The last attribute for the link profile is the Data Rate. DS3 is the speed of data transmission over the link.

3.2.2.4 Base Station and Weather Station Attributes

Base Station and Weather Station attributes are discussed in this section.

Here, the simulation data of some parameters of the two units is discussed.

WiMAX Weather Station “Delay” and “Throughput” parameters have been selected as shown in figure 3.2.2.4.1.

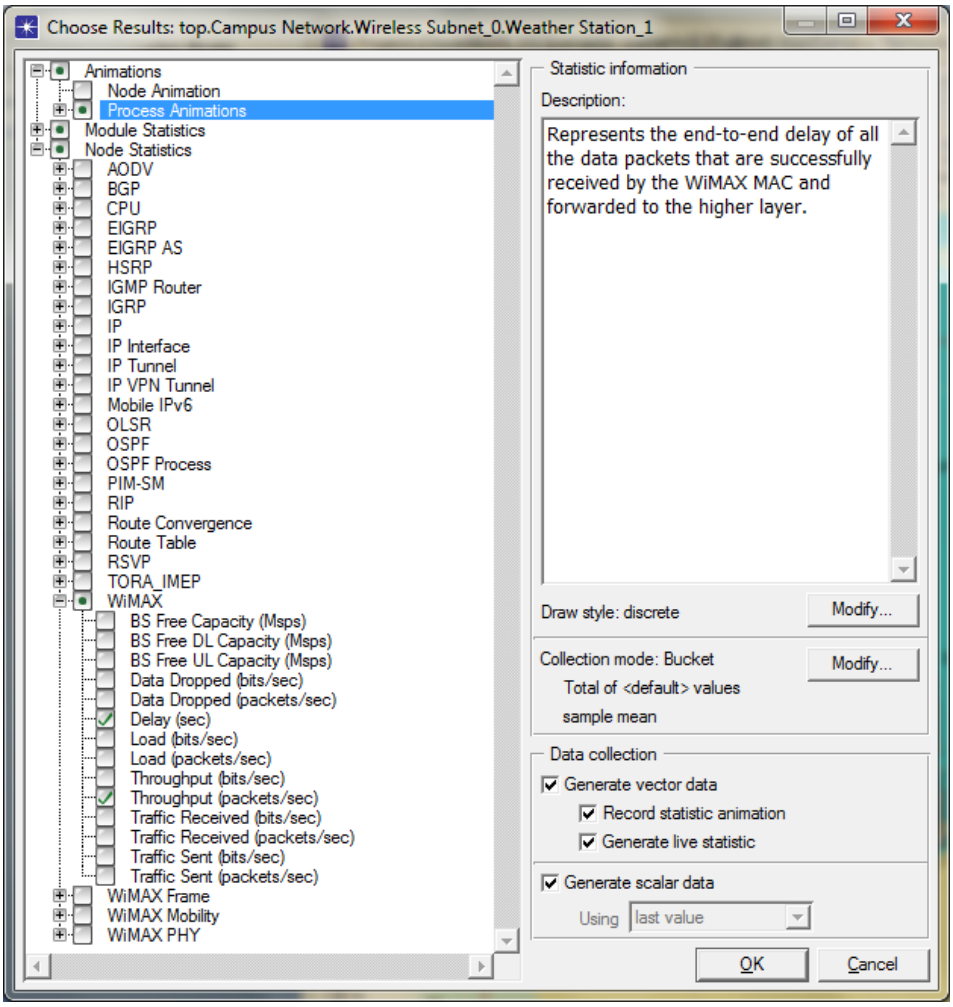


Figure 3.2.2.4.1: Choose Results Window for Weather Station

The “Busy” & “Throughput” parameters for directions for all links have been selected within the appropriate window as shown in figure 3.2.2.4.2.

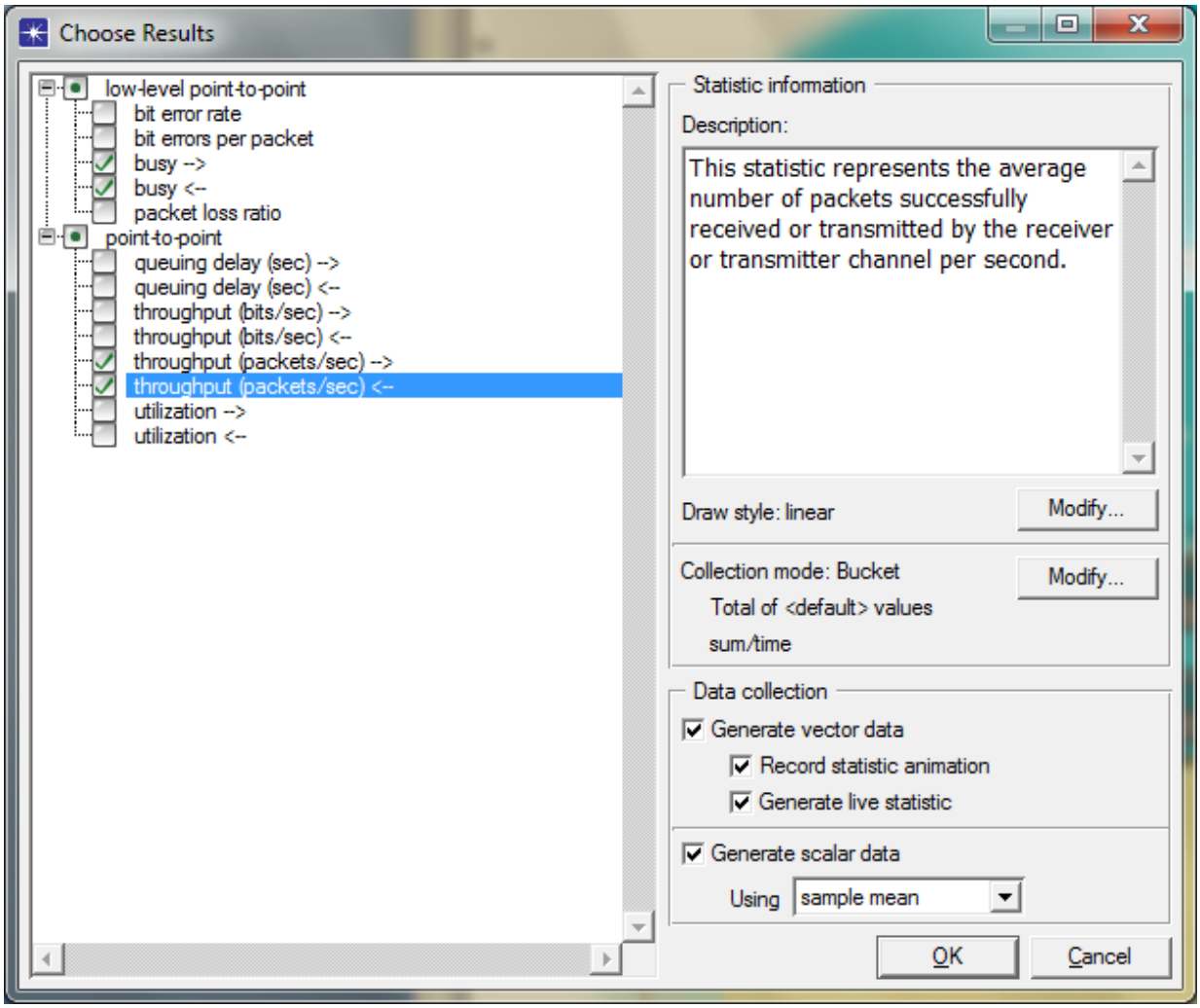


Figure 3.2.2.4.2: The Data for Four Parameters is collected during the Simulation Session.

3.2.2.5 Subscriber Station

The Radio Receiver data is collected during CloneWAN simulation as shown in figure 3.2.2.5.1. The other nodes hat are collecting the data during the simulation are the Voice schemes. If the users are using videos, these nodes would add extra delay.

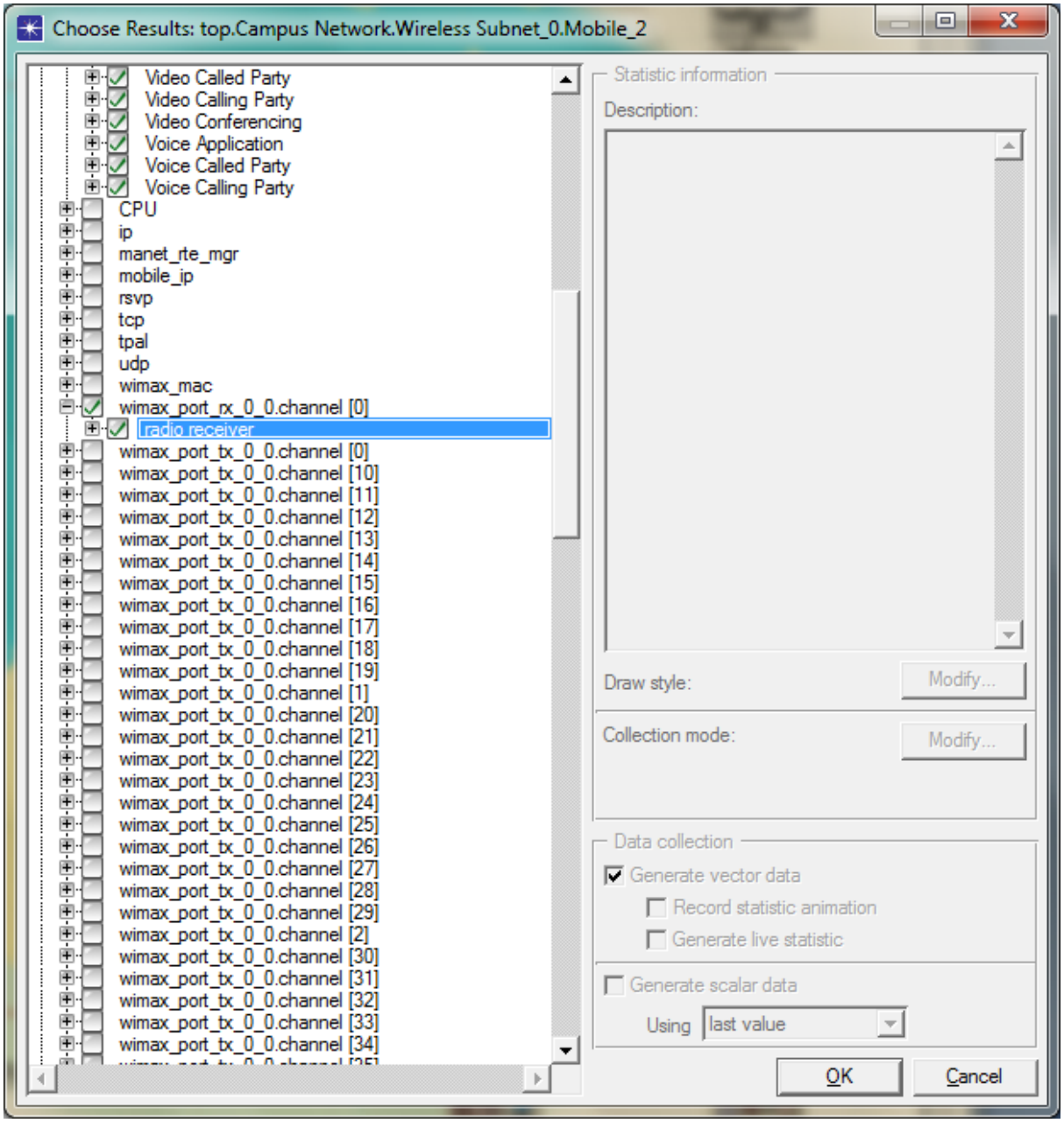


Figure 3.2.2.5.1: Subnet 2 Choose Results.

3.2.2.6 Opnet Aapplication Model Hierarchy

To summarise, Opnet application model hierarchy can be expressed as shown in figure 3.2.2.6.1

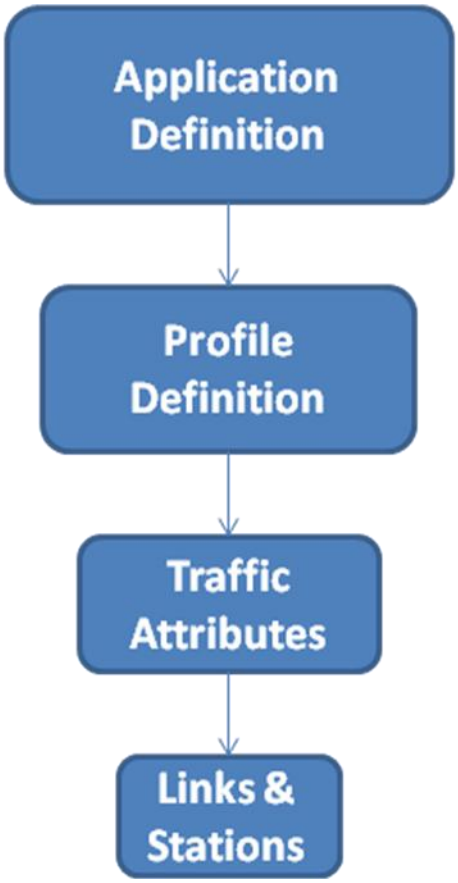


Figure 3.2.2.6.1: Model Hierarchy

The application definition is related to customers applications. For example, they may the network for video only, other customer may is it for Database or mixed applications. The profile is related to the type of users, for example engineers, bank officers or mixed. Traffic attributes is used to define the time of the use such night, day time or weekend or all. The links and stations are associated with the type of the link. There are approximately 50 types of links and WiMAX offer the option to select the one of interest.

3.2.3 Section Summary

Opnet is structured in a hierarchical pattern and if it is followed accordingly, some interesting simulation results are expected. This section provided Opnet application model.

3.3 CloneWAN Simulation Results

The required parameter to measure the performance for our new station, the weather station has been simulated within this section.

3.3.1 CloneWAN Delay and Throughput

Opnet CloneWAN model simulation has been completed and collected results with total of 12,700,821 events in 2min, 28 sec elapsed time for 40 nodes.

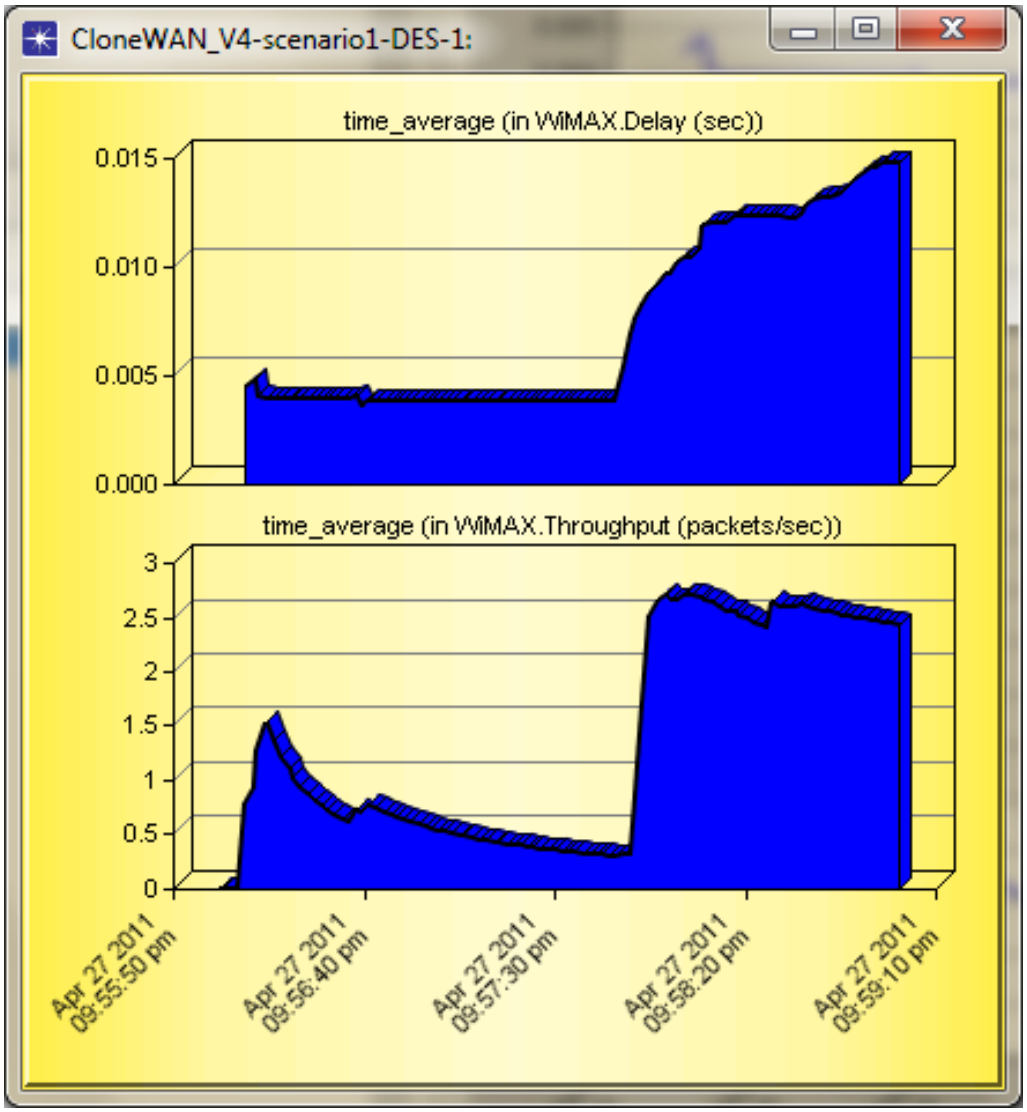


Figure 3.3.1.1: CloneWAN Delay in seconds & Throughput in packets per second

Figure 3.3.1.1 presents CloneWAN delay and throughput parameters. As traffic builds up generally the delay increases and the throughput quality improves. The network is initialised within the first 3.5 seconds. After 3.5 seconds, approximate, the model deals with the subscribers according to our initial setup to the network.

The Traffic for Weather Station received and sent in packets per seconds is shown in figure 3.3.1.2. It shows that there is an increase in activity within the last part of the traffic for both directions. This behaviour is to do with the statistical formulas that Opnet used to simulate a network scenario. This is consistent with the CloneWAN traffic shown in figure 3.3.1.1 above.

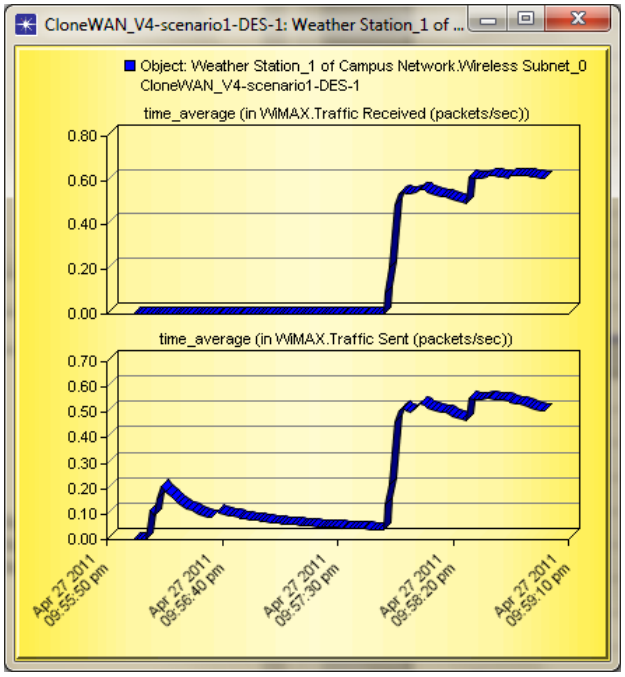


Figure 3.3.1.2: Weather Station Received and Sent Traffic Scenarios

The increase in traffic for the Weather Station is due to heavy use of Http by the Day Users. Figure 3.3.1.3 shows the Http use has increased within the last part of the traffic.

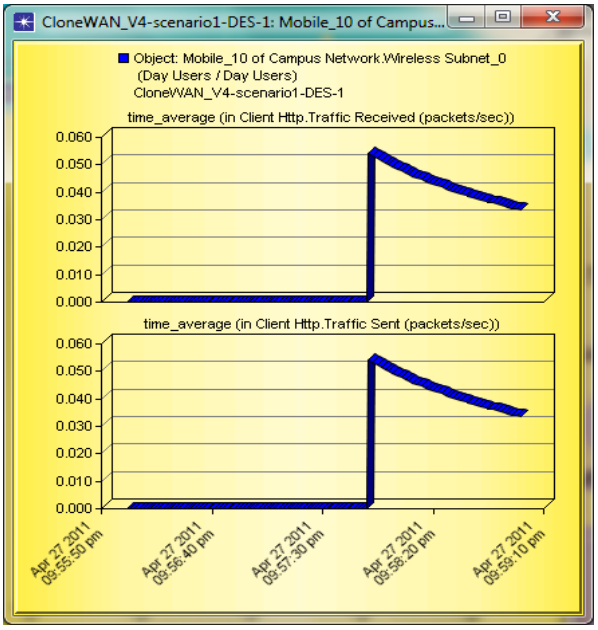


Figure 3.3.1.3: Http Traffic for Weather Station

3.3.2 CloneWAN Diagnose and Prediction

Opnet has built in tools to perform diagnose and predict application behaviour; called "Application Characterization Environment", ACE.

ACE predicted that Weather Station utilization is full as shown in figure 3.3.2.1.

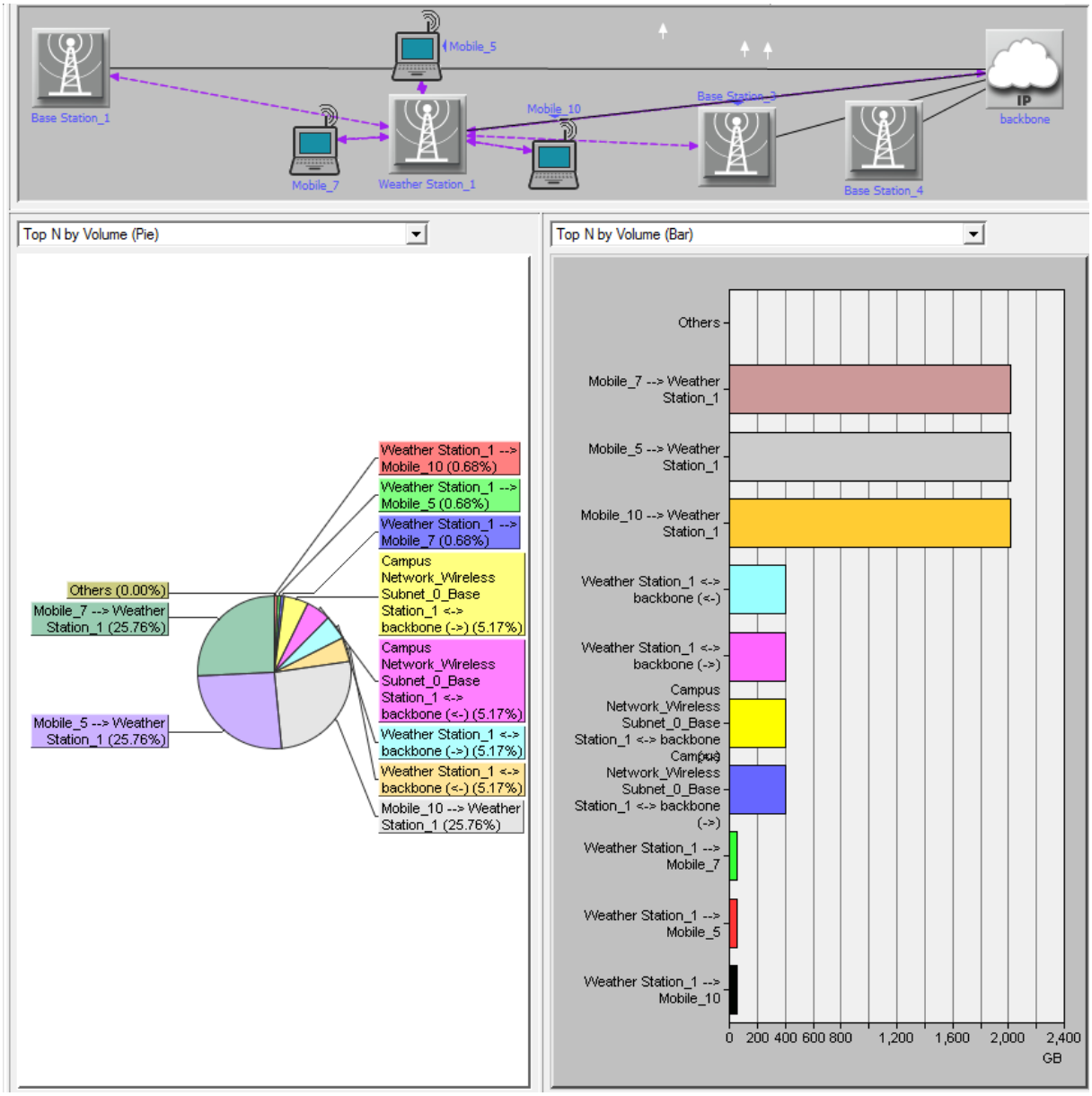


Figure 3.3.2.1: ACE CloneWAN Network Load Distribution by Volume and Bar

Subscribers 5, 7 & 10 to Weather Station are occupying each 25.76% of the traffic and the rest of the traffic is distributed with the other direction of the traffic. As ACE has identified that WeS is more occupied with Ss → WeS, with the next stage of the research, upon completing the Weather Sensor and receive data from the Met Office, CloneWAN should reduce the percentage of usage with this direction to allow Alert Messages to be sent without the possibility of jamming.

ACE provides facility to increase traffic to the network model, this facility called “Add volume to traffic”. Three requests have been placed to increase the traffic. The first one is shown in figure 3.3.2.2.

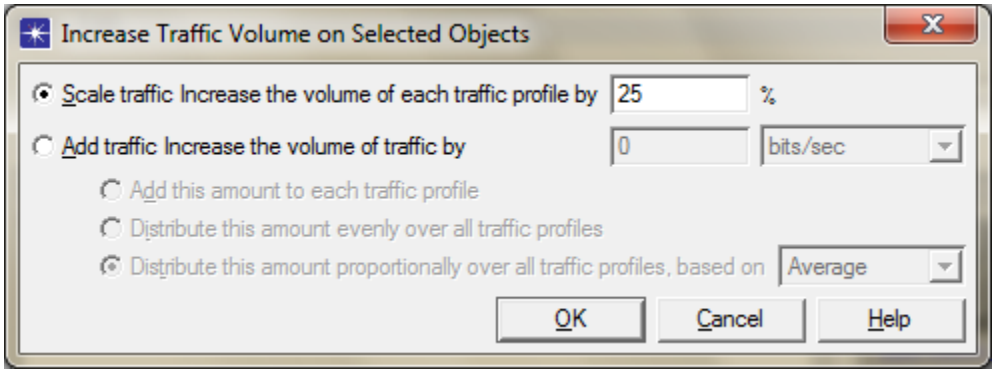


Figure 3.3.2.2: Increase Traffic Volume on CloneWAN

This has been used and requested the traffic profile to be increased by 25%. ACE reported minor changes. Even when requested to increase the traffic up to 75% and 100%, the changes are still minor.

As indicated within this chapter that the simulation duration for CloneWAN is 3 mins. However, another facility has been used here to roll up the duration of this model. Figure 3.3.2.3 shows the Roll Up facility that offered by ACE.

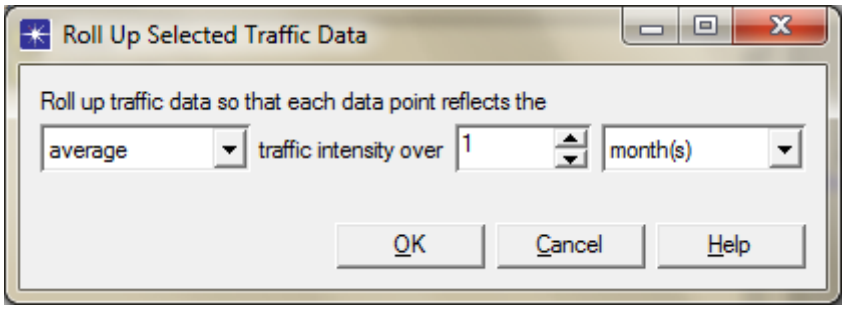


Figure 3.3.2.3: The Roll Up facility

The default setting is 1 day roll up but for the purpose of longer prediction, it is set to one month. The changes to CloneWAN are shown in figure 3.3.2.4.

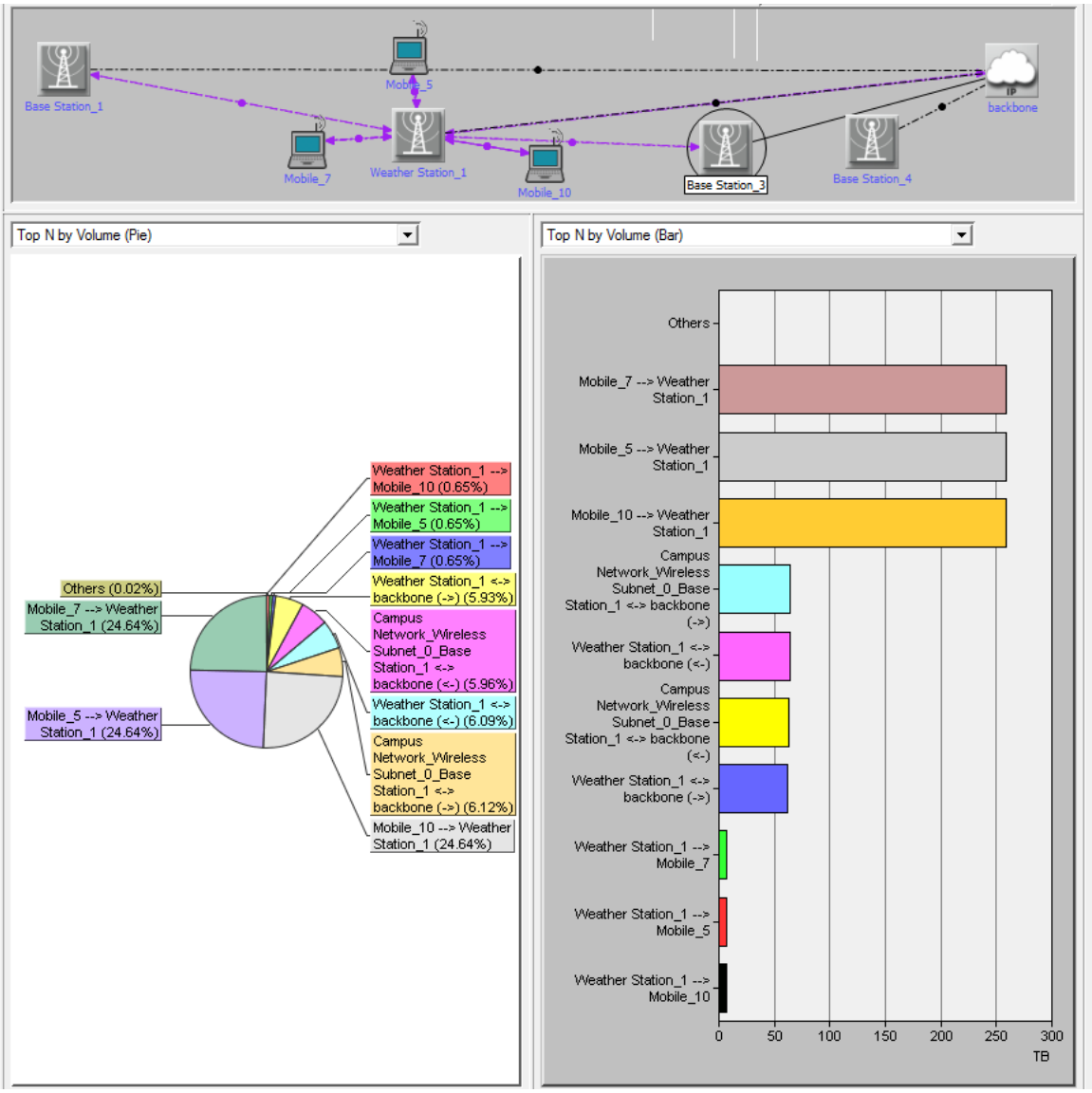


Figure 3.3.2.4: Change of 1.12% if CloneWAN runs for 1, 6, 12 or 24 months with Heavy Traffic

Now, subscribers 5, 7 & 10 to Weather Station are occupying each 24.64% of the traffic, a change of only 1.12%. CloneWAN was rolled up to 6, 12 & 24 months. There are no changes to be reported at all to mean that CloneWAN model is fully configure and working..

By providing values to Baseline and Forecast periods to another tool called Forecast Selected Traffic as shown in figure 3.3.2.5, the tool utilises a regression algorithm that is suitable to the network topology.

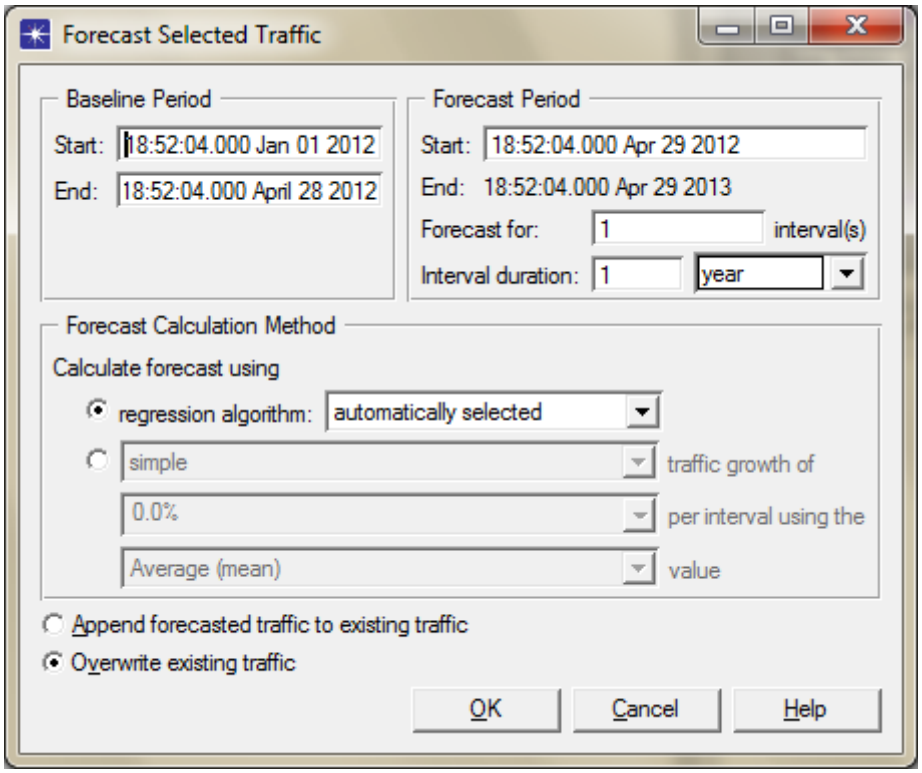
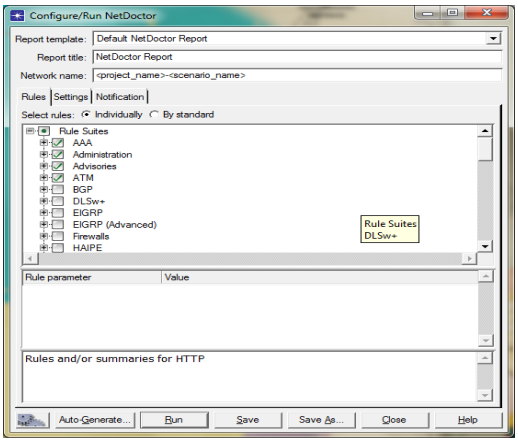


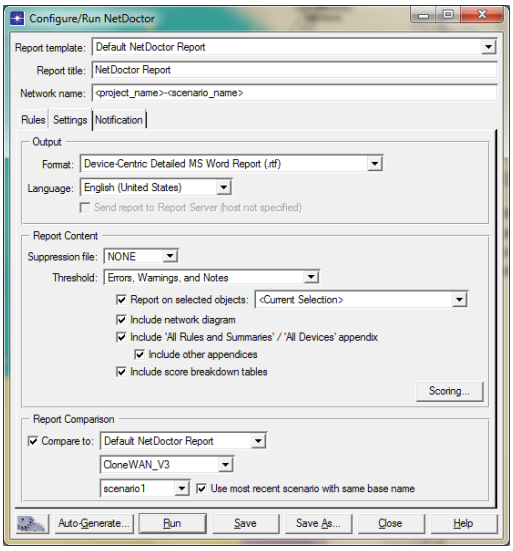
Figure 3.3.2.5: Regression Algorithm is used to forecast the Behaviour pattern of CloneWAN

ACE is predicting that Mobiles 5,7 and 10 will take up 24.64% of the overall network load between 18:52:04.000 April 29 2012 to 18:52:04.000 Aug 23 2013. Hence, there are no changes at all to the current behaviour. The prediction is giving the best distribution of CloneWAN in real practises. This is proves that CloneWAN model is stable network as it is.

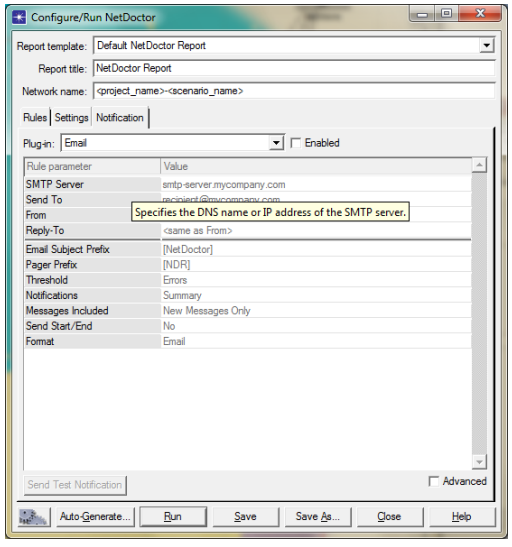
It is worth mentioning that Opnet provides a tool called Net Doctor to notify a Network developer of errors against set of rules and standard. The three windows shown in figure 3.3.2.6 to select the required rules, output documents and notify the user.



(a)



(b)



(c)

Figure 3.3.2.6: a) Rules, b) Settings & c) Notification of NetDoctor

Results of running NetDoctor are in Appendix A. The Executive summary is:

This NetDoctor report shows the state of the network named "**CloneWAN_V6**".The data used to generate this report came from 1 tested device and 154 rules. The score for this report is 100 (out of 100).
No reported issues were found in this network.

3.3.3 Section Summary

The two parameters that are used to evaluate a typical network are Delay and Throughput. This chapter has derived these parameters in section 3.2.1. To simulate a network, Opnet application model hierarchy has to be followed. Step by step, this hierarchy has been furnished in section 3.2.2 Section 3.2.3 presented CloneWAN simulation results while section 3.2.4 diagnosed and predicted the pattern behavior of CloneWAN up to 2013. All shows that the current version of CloneWAN is stable.

3.4 Weather Station (WeS) Hardware Implementation

WeS is based on WiMAX standard base station and DSP with set of sensors. The number of sensors is associated with the number of channels offered by the AD converter.

Figure 3.4.1 shows the overall hardware implementation of Weather Station (WeS).

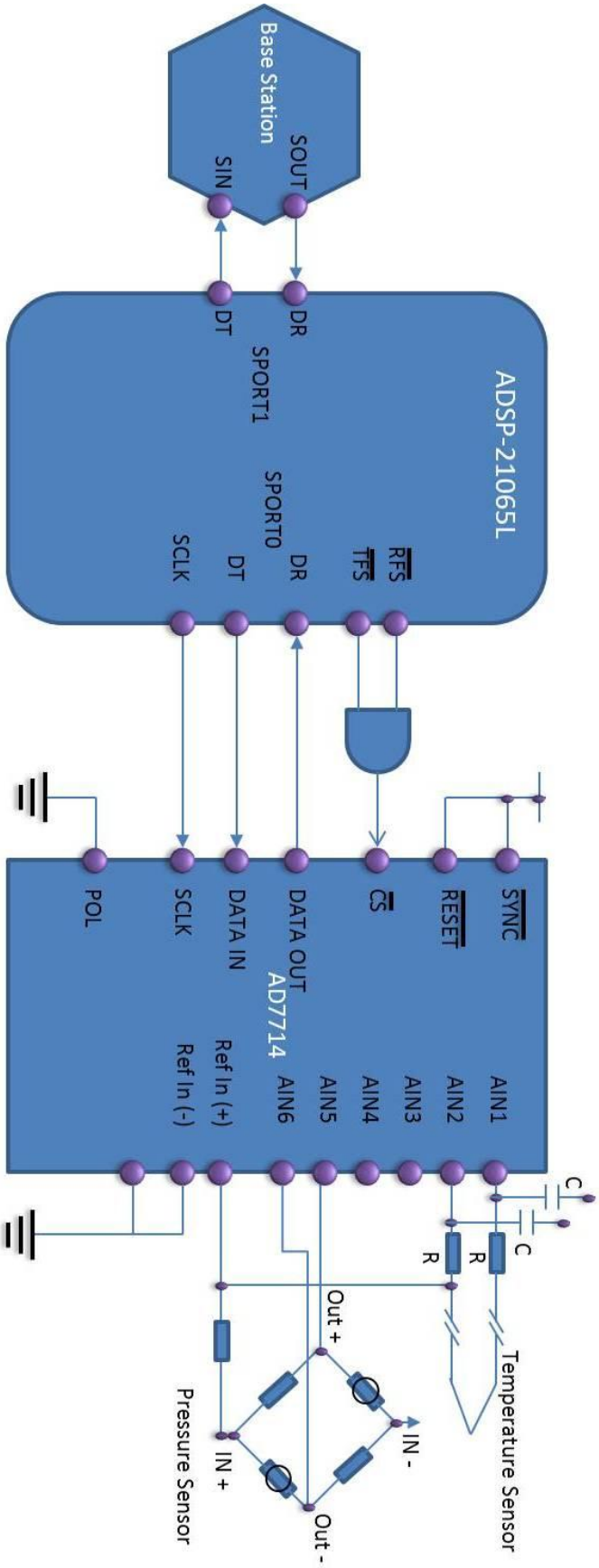


Figure 3.4.1: The overall hardware implementation of Weather Station (WeS)

The interface circuit comprises of three main components:

- 6 Channels AD converter
- DSP
- Base Station

The chosen AD converter is AD7714. The data sheets for AD7714 (AD7714, 1998) provides the pin configuration with Analog Devices DSP processor.

It is suggested that ADSP-21065L to be the processor for this interface but less performance DSP processor will do. The reason for this choice is because of the availability of the hardware locally.

Figure 3.4.1 shows an interface between the AD7714 and the ADSP-21065L DSP processor. The *RFS* and *TFS* pins of the ADSP-21065L are configured as active low outputs and the ADSP-21065L serial clock line, SCLK, is also configured as an output. The POL pin of the AD7714 is hard-wired low. Because the SCLK from the ADSP-21065L is a continuous clock, the *CS* of the AD7714 must be used to gate off the clock once the transfer is complete. The *CS* for the AD7714 is active when both the *RFS* and *TFS* outputs from the ADSP-21065L are active. The serial clock rate on the ADSP-21065L should be limited to 3MHz to ensure correct operation with the AD7714.

The temperature measurement is configured in figure 3.4.1 outlines a connection from a thermocouple to the AD7714.

At the same figure, figure 3.4.1 it shows the pressure measurement connection. Though the figure shows a pressure transducer arranged in a bridge network and gives a differential output voltage between its OUT(+) and OUT(–) terminals, but alternatives can be swapped to the current suggestions. With rated full-scale pressure (in this case 300 mmHg) according to the data sheet, on the transducer, the differential output voltage is 3 mV/Volt of the input voltage (i.e., the voltage between its IN(+) and IN(–) terminals).

The base station shown in figure 3.4.1 is the standard WiMAX base station. The AD7714 to ADSP-21065L Interface shows that the temperature or pressure samples are received to the DSP via DATAOUT (AD Converter) / DR (DSP) via one serial port. After processing the sensor data and formatting them, DT of the 2nd serial port of the DSP transmits the data to the Base Station via Sin pin.

The software model for the weather base station, is described in chapter 4, has been designed to receive the weather information and processes them according to a lookup table.

3.5 Chapter Summary

This chapter introduced a new network technology architecture that is compatible with IEEE 802.16e standard and labelled as CloneWAN. To verify it's validity, this architecture is simulated on the UAE and the results showed full satisfaction between the 7 cells cross the UAE.

In addition, the hardware for the Weather Station (WeS) that includes two types of sensors, Digital Signal Processor and Analog to Digital converter has been designed, presented and discussed in section 3.4.

Chapter four: Weather Station Model

In this chapter, the basic aim and objectives of the Weather Station (WeS) model have been defined. After that the overview of WeS model is discussed. Moreover, this chapter describes the disaster recovery case studies, motivation and then the structure of this chapter.

4.1 Aim

The aim is to implement WiMAX technology with an interface control information which is the built in facility of Opnet 17.0 in order to deploy a wireless network that will keep the template as an early warning message in case of abrupt change in the weather.

4.2 Objectives

The developed WeS model has followed the following objectives:

- To investigate the concepts of 3G and 4G technology.
- To investigate the Characteristics and features of WI-MAX technology.
- To use the interface control information (ICI) in Opnet17.0.
- To implement the ICI with WiMAX using the simulator Opnet 17.0.
- To model the weather sensors and use with WiMAX.
- Designing the algorithm using the proto-c language that will allow the network to behave as an early warning system for any change in weather that is not normal.
- Develop a model by designing the code at the sender side which is the WiMAX base station coupled with sensor.
- Develop a model by designing the code at the receiver side which is the subscriber side in the WiMAX base station in order to receive the interrupt message.

4.3 Overview of WeS Model

The model would require the use of C and C++ to complete an interface between WiMAX base station and an electronic sensor. The sensor is interfaced with a DSP based board to sample and collate the weather information be it from Met office or local and actual data via a sensor. This interface has been described in Chapter 3. Network protocols such as CSMA/CA, Media access control protocol, PKM (privacy key management protocol), Simple network management protocol (SNMP), are required to complete the interface. The software used for the purpose of designing weather station is Opnet modeller 17.0. The design

methodology adopted in order to complete the interface is ICI format [5] which is discussed in detail in section 4.7.

4.4 Motivation of WeS Model

The main reason of the motivation of this project is the deployment of a system that will behave as early warning system which sense hazard conditions like earthquake, tsunami and the severe weather conditions and so on. Soon the system sense the extreme conditions, it automatically generates a message and forward it through the base station, which further forward that message to the subscribers.

WiMAX technology is being used in the project because of its high data rate in environments like point to multipoint, LOS (Line Of Sight) & over long distance having high broadband connection and above all it is IEEE standard (IEEE, 2004). WiMAX is a growing leading technology in the market because having the same services to WIFI but having the capability of covering more area and good quality of service. The ICI is used for this project due to its benefit of ease to design. ICI actually sends the particular format along with packet to the broadcasting device to control a particular event.

The design of WiMAX model with ICI can be implemented through simulation tool known as Optimum Network (Opnet) of version 17. Opnet is an object oriented based tool for simulation that creates a visualized simulation environment for the network modelling. It has the ability to examine the behaviour of all networks. The software has layers that form a hierarchical architecture. This hierarchical structure can be divided into three domains they are network domain, node domain and process domain. For network topologies, geographical coordinates, sub networks one can use network domain. Node domain is used for single network nodes that includes queues, processor routers etc. Finally process domain is for programming e.g. source code inside the network nodes.

4.5 WiMAX with ICI Format

As discussed earlier that the purpose of the chapter, among other items, is to understand WiMAX technology, its operations and extended with the implementation of WiMAX with ICI format using Opnet. It is essential to be familiar with the Modeller Optimum Network (OPNET) 17.0.

4.6 Natural Disasters, WiMAX Technology (IEEE 802.16e) and OPNET

WiMAX is a wide technology that meets the requirement of the modern world in a sense of wide coverage, high data rates, and is economical but the characteristics of this wide technology can be explored using the research based software Opnet which is powerful and feasible for this technology as this software provides the complete package of WiMAX to

develop, design and test new technologies. For the disaster management one requires the advanced wireless sensor networks that are possible with the help of only two technologies that is LTE and WiMAX. Wi-Fi cannot work because of its limited coverage for disaster management one needs the wide coverage network through which information can be spread quickly. LTE is costly as it requires more antennas at the base station while WiMAX is less costly for this disaster management (Fazel, 2008).

4.6.1 Natural Disasters

Lately, the world has encountered cases that are labelled as disasters.

Disasters covered wide areas of the world, from China to Pakistan, to Middle East, etc. In 2005 Pakistan encountered many natural disasters in which earthquake in 2005 at this placed caused serious damages to infrastructure of many parts of the country like Azad Kashmir and Muzaffarabad (SDMA, 2010). In Saudi Arabia there was a flood in Jeddah in 2010 causing destruction of building, infrastructure even human lives. These disasters left the population without water, electricity, food, and telecommunication, the task for reconstruction, rehabilitation of the community and people were facing very tough challenge to victims. Hence, the need for deploying a sophisticated network system is becoming a vital requirement for the survival of the people. These disasters raised the concern of deploying an advanced technology for early warning systems to save and reduce the loss of many lives of people and animals. According to the latest research in science and technology deploying WiMAX technology in these areas is being considered seriously as a most convenient applicable solution to the natural disasters (WiMAX-Communication, 2011).

4.6.2 Historical Background

WiMAX is becoming popular among the latest technology of the world. As the Opnet is based on the discrete event simulation this prominent feature of Opnet helps in analysing the performance of WiMAX network. Research work is done on the WiMAX network using Opnet by modelling the different architectures that give assistance to bandwidth and delay, Quality of service guarantees for IEEE 802.16 networks. A request has been made in various research papers to explore the two important layers that is physical layer and MAC layer to improve the system performance and security (Nauymi, 2007). It is realized that for the small cells with low power base station called the femtocells WiMAX is the best option (Deruyck, 2010). This evidence is revealed using Opnet when comparing the three different technologies WiFi, WIMAX and UMTS. The results describes that WiMAX is the superior technology as compared to other two WiFi and UMTS because WiMAX throughput is high and this technology outperforms not only in sense of throughput also including delay and therefore WiMAX is preferable in comparison with other two air interfaces. In the latest

research it is shown that WiMAX throughput is more than that of LTE Life (Guangzhou, 2009) (Zhang, 2008).

4.6.3 Introduction to 3G

The technology actually becomes known for the communicating systems like for example cellular communication systems. The technology is introduced to meet the high data rate requirements (Guangzhou, 2009). The example of 3G technology is the WCDMA. The chipping code is used to encode information and hence data rate 3.84Mbps. There are many applications of this technology. It is used in VOIP, video on demand, video conferencing and mobile TVs etc. High data rates and security is the superiority of 3G over 2G. 3G unable to meet requirements of some new applications so experts started thinking of new technology to meet new application requirements.

4.6.4 Introduction to 4G

Currently 4G technology is growing and leading the market. Examples of 4G technologies are LTE and WiMAX. They are desired because of their unique characteristics for example the modulation technique is advanced that is OFDM and they operate on high frequencies and hence high data rates.

4.6.5 WiMAX

WiMAX is a broadband wireless network technology that is IP based and is providing quick connections. It provides point to multi point broadband access. WiMAX is the abbreviation of Worldwide Interoperability for Microwave Access (WiMAX). The common phrase used for WiMAX technology is the ability to provide the broadband access up to the last mile that means it is such a unique technology that is capable of providing network connections up to the last end from service provider to subscriber. In other words it can be said that it has wider range to cover all corners in providing connectivity. One more thing about WiMAX is noticeable that its operation is on both frequencies licensed and non-licensed. WiMAX uses licensed bands of 2.3, 2.5 and 3.5 GHz. A set of standards known as IMT-2000 was introduced in that technology, which is defined by international Telecommunication Union, so if any one country uses the equipment of WiMAX if they recognize that standard (Flickenger et al., 2007).

4.6.5.1 WiMAX data rates and modulation technique

WiMAX is provided with high bandwidth and since bandwidth is directly proportional to data rate, it means that the more the bandwidth the more the data rate. This technology provides the high data rate of 75 Mbps. As the data rate is linked to network access range, therefore because of high data rate it provides the range up to 75 km (Nauymi, 2007).

As in each technology the modulation technique used is different .WiMAX use OFDM modulation technique. Actually high data rate in WiMAX is possible because of this modulation technique. This modulation technique does not use single carrier there are a lot of carriers used in this technique 256 sub-carriers that actually increases its performance and it outperforms other technologies.

4.6.5.2 WiMAX Physical Layer Features

There are three types of features of physical layer. They are packet loss modelling, PHY layer Overhead and Impairment Modelling and the last feature is partial usage of subchannels. These features are described briefly one by one.

1) **Packet loss Modelling**

Packet losses experienced by the network can be recorded by using the physical modelling. It can be observed the packet losses because of physical layer effects.

2) **PHY Layer Overhead and Impairment Modelling**

This model includes a lot of effects for example PHY Layer overheads, co-channel interference for SOFDMA, Pathloss, multipath fading for SOFDMA channels.

3) **Partial use of sub channels**

This feature is used in the WIMAX model to reduce the interference. It permits to split the sub channels into disjoint set of sub channels.

4.6.5.3 WiMAX MAC Messages and Signals

This section includes the MAC Data Messages, MAC Control Messages, and MAC Control Signals.

The example of MAC data messages includes MAC PDU where PDU is the packet data unit. In MAC control messages a lot of messages are supported. For example of some messages are ranging request, ranging response, ARQ Discard message, ARQ Reset message, scanning interval allocation request, scanning interval allocation response etc. The last type of signals supported are the MAC control signals examples of these types of signals CDMA code which is used in the bandwidth request systems. Another example is the CQICH code word which is used to operate the adaptive code and modulation decisions from MS to BS.

4.6.5.4 WiMAX Mobility Features

This covers the Hand off, Ranging and initial SS-BS Association, Adaptive Modulation and Coding, Open Loop Power Control for uplink channels and Access Service Network Messages. These features are discussed one by one.

1) **Handover**

Handover features include modelling of Neighbour Advertisements, Scanning which can be initiated by the mobile subscriber or base station, hard handover etc.

2) **Ranging and initial SS-BS Association**

The model provides the ranging like CDMA. Efficiency mode is set to mobility and ranging enabled then initially connection between SS and BS is controlled by initial ranging. As SS moves scanning are used for the same purpose.

When any other efficiency mode is used the attribute **WiMAX parameters > SS parameters >BS MAC address** is used for SS and BS connection.

4.6.5.5 Mobility features that control SS (subscriber stations)

1) **Adaptive Modelling and Coding**

This feature is enabled by default. The advantage of using this is that higher data rates transmission occurs hence robust transmission is possible due AMC (Adaptive Modelling and Coding).

2) **Open Loop Power Control for Uplink Channels**

This feature plays a vital role in power adjustment it enables the SS to use the optimum power and hence reducing the power consumption and inter-cell interference.

3) **Access service Network Messages**

ASN provides the means to connect mobile subscribers to an IP backbone with session continuity. ASN comprises base stations (BS) and access gateways, called ASN-GW. The interface between the ASN and mobile subscriber is through BS with IEEE 802.16e.

4.6.5.6 WiMAX MAC Layer Features

MAC layer features include quality of service .The MAC layer features make it possible to have the quality of service. Various factors include obtaining the quality of service for example it includes the concept of buffering and queuing. There are many types of

scheduling services supported by this layer UGS, nrtPS etc. because of this feature BS is able to distribute uplink and downlink transmissions of PDUs.

There are two types of code schemes that are supported by this layer it includes the convolutional turbo coding and convolutional coding. WIMAX works with both IPV4 and IPV6.It gives assistance to broadcast and multicast traffic when the physical layer is at work.

Reserved sub frame capacity is another prominent feature supported.
WiMAX Parameters > BS Parameters > Reserved DL Sub frame Capacity.

Some bandwidth parameters are supported. It includes CDMA contention based requests, Piggybacked bandwidth requests.
WiMAX Parameters > SS Parameters > Piggyback BW Request.

4.6.6 Introduction to OPNET Proto C

OPNET is a network simulation tool. It allows the user to use its library with a wide range of functions. It has tools that include for example designing a model, performing the simulation and result analysis. Any type of network wired, wireless, LAN, WLAN, WiMAX, LTE and satellite networks can be built within Opnet. All types of network that are linked to each other can be simulated. OPNET users can therefore assign attribute settings and observe the impact of the settings. Utilizations, Delay, Traffic sent per second, Data message flows; packet losses, link failures, bit errors and a lot of other parameters can be viewed.

4.6.6.1 WiMAX Topologies using OPNET

Opnet provides a wide range of facilities different topologies or scenarios can be made with ease .WiMAX Topologies are shown in the figure below.

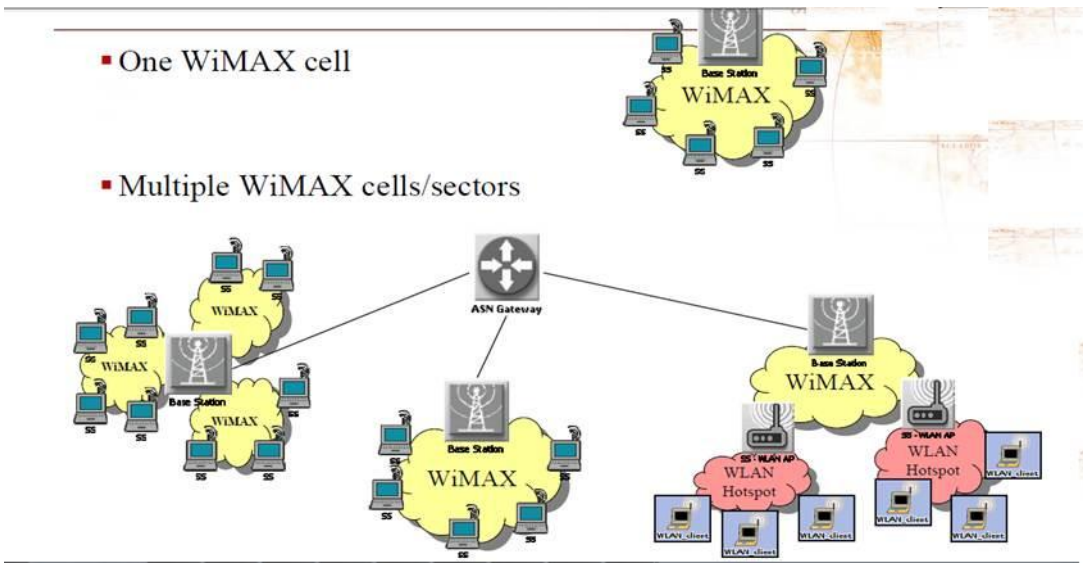


Figure 4.6.6.1.1: WiMAX Topologies

Handover in the WIMAX is shown in the figure below.

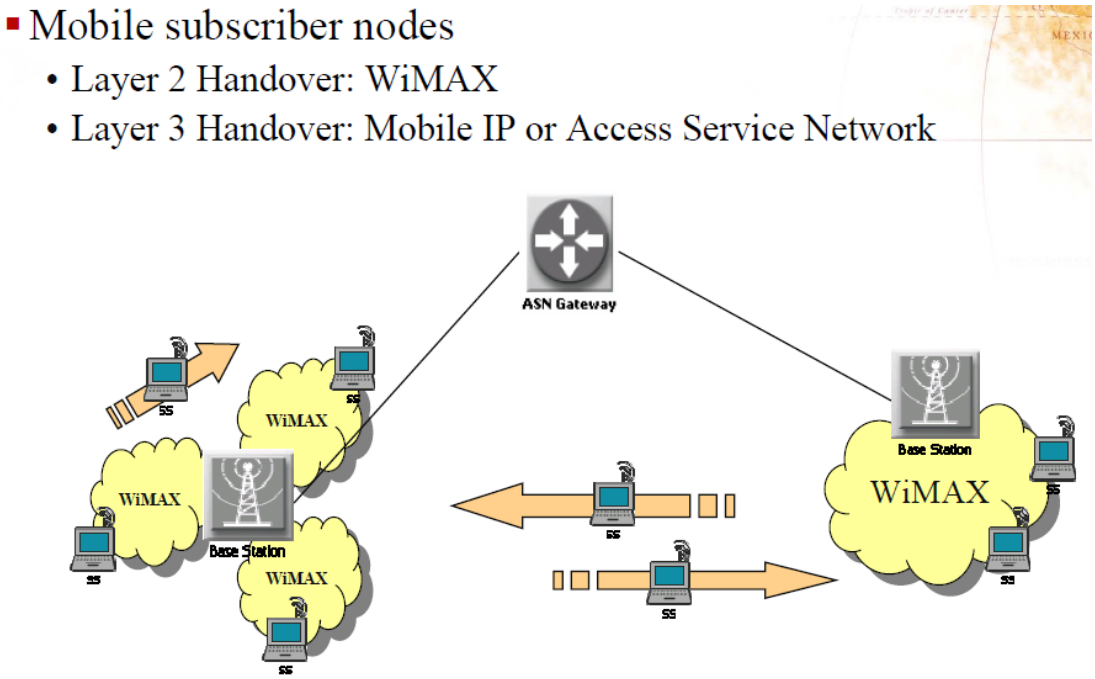


Figure 4.6.6.1.2: Handover in the WiMAX

The example in figure 4.6.6.1.2 shows two BSs and a Gateway. The handover runs at two layers. Layer2 is related to WiMAX to WiMAX handover and Layer3 is from the Gateway to WiMAX BSs.

4.6.6.2 WiMAX Model Entities in Opnet

The following figure shows how one can make the WiMAX configure accordingly what is desired.

In the work station model there are options to adjust the Antenna gain, transmission power and many more. Similarly in WiMAX base station there are parameters that one can adjust for example routing protocols, VPN, System management, WiMAX BS parameters and many more and in the same way WiMAX config have the parameter like channel coding, contention parameters and Efficiency enabled.

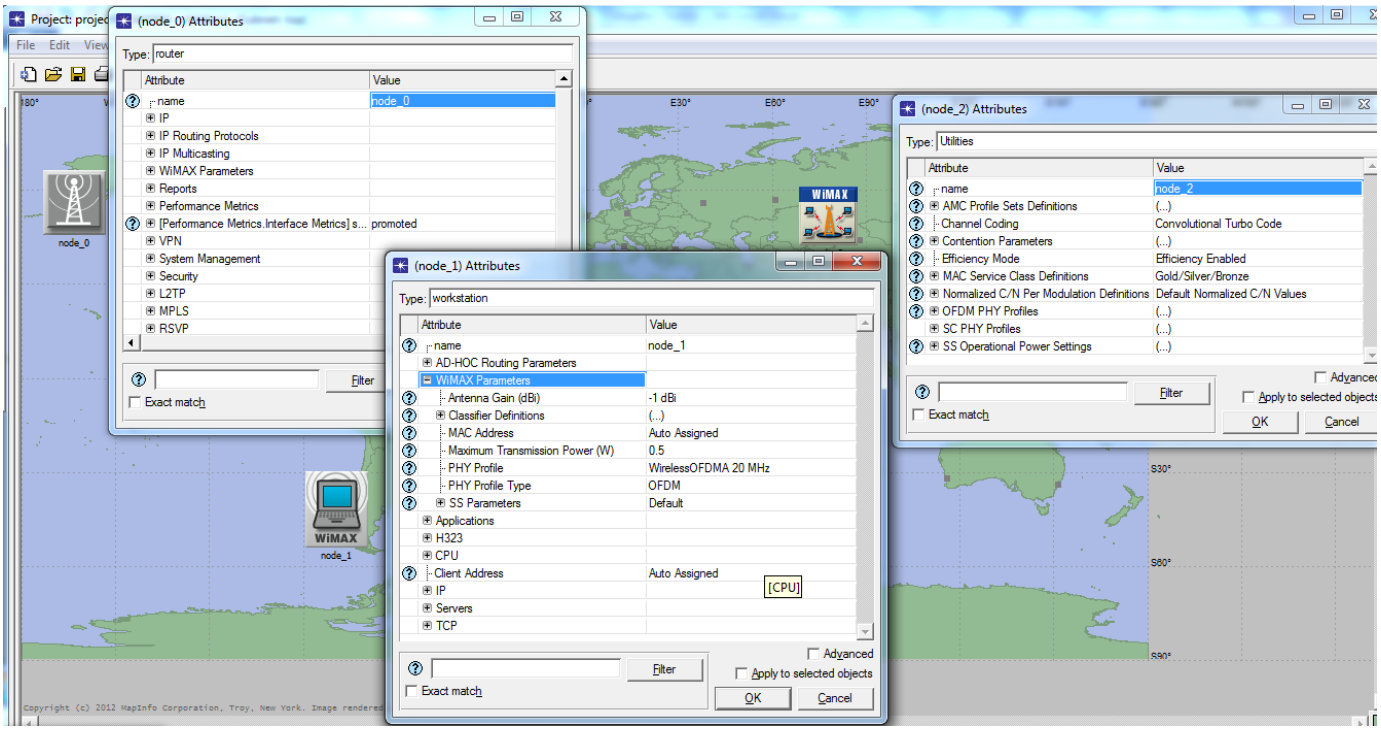


Figure 4.6.6.2.1: WiMAX Model Entities

4.6.6.3 WiMAX Model Abstractions in Opnet

In WiMAX config one can have the four levels of abstractions in Efficiency mode. They are defined below one by one.

1) Efficiency Enabled

This mode does not involve the function of physical layer it deals with MAC layer. This mode is enabled when capacity planning is major focus of the network design.

2) Framing Module Enabled

This mode includes both physical layer and MAC layer functions. However, this abstraction is used for quality of service for the given network. Framing Module Enabled provides more accurate delay than the first case.

3) Physical Layer Enabled

It deals with the channel effects, co channel Interference, multi path fading, Path loss effects and broadcast connections.

4) Mobility and Ranging Enabled

This last case is used for mobility modelling, scanning and hand over delays. It also deals with initial and periodic ranging that includes delays and MS power levels.

4.6.6.4 Section Summary

The background on WiMAX and Opnet has been covered thoroughly in this section. WiMAX technology is a wireless wide range network access technology. It is getting more popular for its high speed and data rate features. The working of WiMAX is described explicitly here. The details of WiMAX extended to cover the OPNET environment and hierarchical layer architecture is presented. Furthermore in order to understand the working of OPNET the WiMAX model entities have been shown. The section covers the detailed range of WiMAX possibilities provided by Opnet.

In the next section, ICI format, code explanation and implementation of the Project are described in detail in order to achieve the goal of the project.

4.7 Design and Implementation of the WeS Model

Having discussed the specifications of WiMAX and presented the heretical structure of Opnet in the previous section, this section discusses the design and implementation of the WeS Model.

4.7.1 Design and implementation of the Project

The previous section provided the background on WiMAX, and Opnet and their benefits. The working of WiMAX and OPNET is described separately. Here, the design implementation is covered and discussed fully for this project. Firstly, the steps are defined how one can create and simulate custom model. Secondly, the node model design process model design along with suitable static variables and temporary variables are included that are the essential requirement of the code. The design of process model is extended with the code explanation in the sender side and the receiving side and handling process is described. The next section of this chapter is supported by the Opnet design methodology that is extended with possible design approaches (two) that one can have in order to complete the project. However, the project is completed using ICI approach which is described in detail in this project.

4.7.2 Creating custom models using OPNET APIs

This section explains how one can build and program custom models in Opnet with advanced case studies. The benefit of this section is that it helps reader to lay foundation of knowledge on custom model creation. Therefore this section covers the techniques that are required for custom model creation, model optimization, simulation, results visualization, comparison and analysis.

4.7.3 Steps for creating custom models

The following list of steps is for building custom models:

- ✓ Design the process models in the process domain using forced and unforced states and state transition diagrams.
- ✓ Implement the node model that is the custom node in the node domain using packet streams and processor modules.
- ✓ Optimize and validate models.
- ✓ Compile and debug process models.

The following list of steps is that are essential for simulating custom models.

- ✓ Create new project in the project domain, name it and design scenario in this project.
- ✓ Design the network topology for the given scenario and check the built links are correct for the designed network.
- ✓ Choose the statistics that are desired.
- ✓ Run simulation for the designed network few times.
- ✓ View, compare and analyse statistic results.

It is not essential to accomplish the above steps in the given order.

4.7.4 Custom Node Model

WeS model has to be consist of receive and transmit modules. WeS model using two processor modules, one transmitter module, one receiver module and packet streams to connect modules is shown in figure 4.7.4.1, Weather Station Node Model.

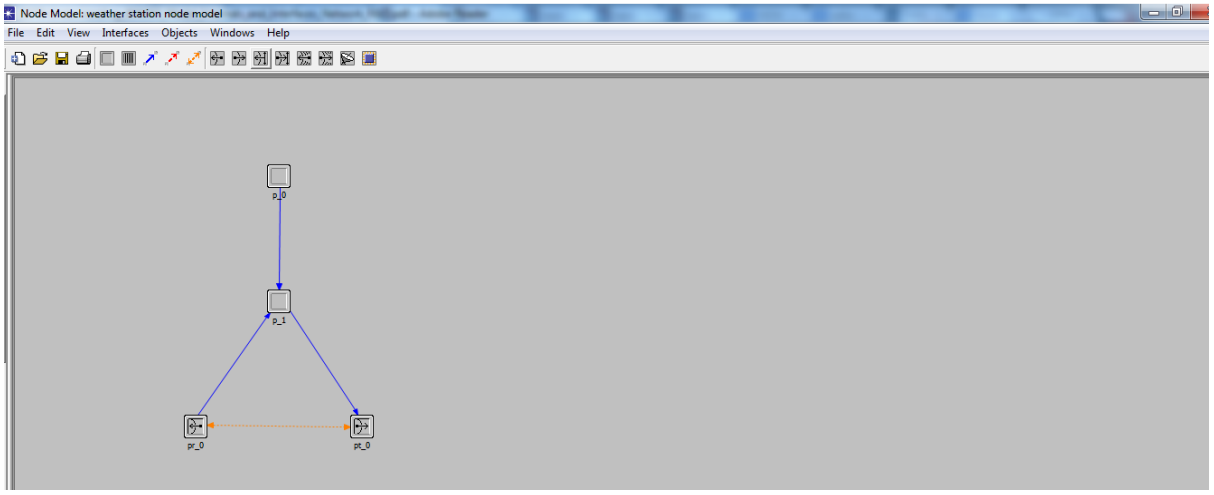


Figure 4.7.4.1: Weather station Node Model

4.7.5 Mechanism of Node Model

The directions of Opnet modules are set up by editing the attributes of each node. Hence:

- The attribute of the first processor module is set as simple source
- The packet inter arrival time has been set as exponential. This attributes enables the model to generate traffic with exponential packet inter arrival times.

Second processor is attached to the first processor so that this processor is able to forward packets to the transmitter and destroyed the packets from the receiver.

- The attributes of this processor similar to the above but the process model of this processor needs to be designed so that the process model for this processor would not be simple source.

The above steps have been achieved sequentially and successfully.

4.7.6 Process model

The process model for the second processor module is shown in figure 4.7.6.1

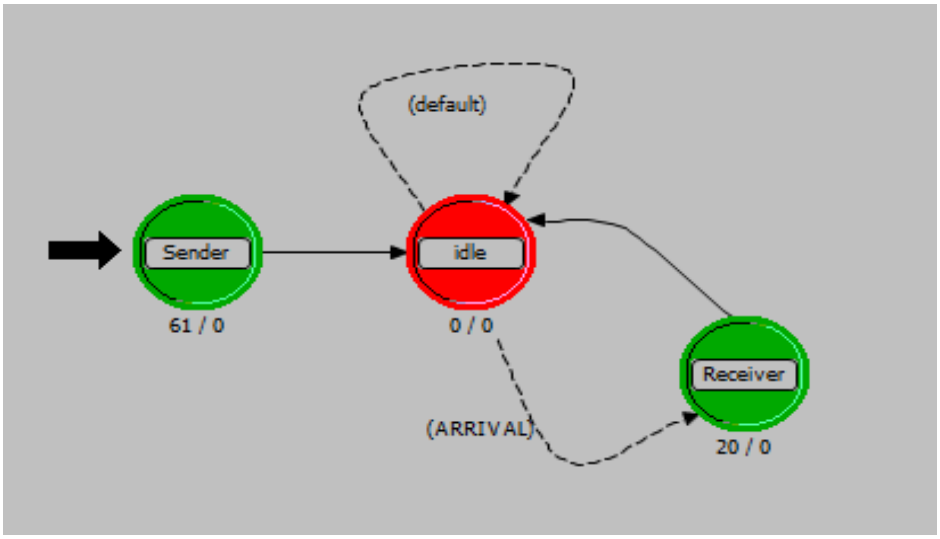


Figure 4.7.6.10: Weather Station Process Model

WeS process model consists of three states. The first one is the Sender, the second one is idle states and the third one the Recover. Any packets arrives must visit the Sender state and without conditions, the packet will land to the idle state to activate the implemented interrupt(s). If the condition of the direct path between idle and Receiver states is true, the packet will be treated by the Receiver state.

4.7.7 Mechanism of process model

The logic flow diagram can be described as follows. The process model shows the logical behaviour of entities, modules, the code will execute the sequence according to the states. The program will wait for the new interrupt to occur and that is why it is generally said a logic flow diagram. It is described in the paragraph given below.

The control initially enters the sender state, and executes the code in the sender state and then it transits to the idle state. At the idle state control simply waits for the UPPER_STRM and LOWER_STRM condition to be triggered. Once the condition is triggered the control moves from the idle state to receiver state and then transition back to the idle state to wait for the new interrupt to come. This process model will repeat itself. The Sender State and the Receiver State are described below.

4.7.7.1 Sender State

In sender state, a code has been designed and added to load a uniform distribution between integer 0 and 40. The coding is done in order to get the object node object ID and its value is stored in the node id state variable. This will generate the integers randomly.

4.7.7.2 Receiver State

In the receiver state, the following tasks have been implemented:

1) At the receiver side the ICI that is linked with the packet has been applied.

2) After that the information of the ICI is copied in the data structure.

3) Finally, ICI content have been extracted or read out so that they can be discarded afterwards.

The next stage is to design code to link the interface control information object with packets and to reach interface control information objects. Finally, the process model has been opened and appropriate static variables and temporary variables in static and temporary block have been declared respectively. At the same level, code in the receiver state has been added. The process model has been compiled. There was no syntax error in the process model at this stage. The following is the code example:

```
wrapper_pk = op_pk_get(LOWER_IN_STRM_INDEX);

ici = op_pk_ici_get(wrapper_pk);

op_ici_attr_get_int32(ici, "id", &temp);

sprintf(msg, "ICI id: %d", temp);
```

```
op_sim_message("ICI test", msg);

op_ici_destroy(ici);

op_pk_nfd_get(wrapper_pk, "header", &header);

op_pk_nfd_get(wrapper_pk, "payload", &pk);

if(header > 39)

        print_pk_size(pk);

else if(header == 30)

        print_pk_size(pk);

op_pk_destroy(pk);

op_pk_destroy(wrapper_pk);


++pk_count;

op_pk_destroy (op_pk_get (op_intrpt_strm ()));

op_stat_write (pk_cnt_stathandle, pk_count);
```

4.7.8 Design Methodology1

Opnet has the other powerful feature that is the ICI format. ICI is the interface control information Package to handle the ICI format object and connect it with packet. It is a kind of data type in the Opnet modeller. It has the wide range of supporting many types of data for example integer, double and structure type of data.

4.7.8.1 ICI based Design

ICI is used to carry out the addition information with packets. Interface control information can be attached with the packet in order to make the packet enable to carry additional control information.

In the ICI format two attributes are defined. However one can define more depending on what is required to design. For weather station design two attributes are designed that is the temperature and humidity. Humidity value id defined 40 and temperature value is -60 to 60 and this information is attached with the packet. These are the particular values on these

values the network will trigger and send this information to user that weather is not normal and some threat may occur.

However one can use the header field to carry information with the packet and the above procedure (the procedure described in this section 4.7.8.1) can be designed by using the header field to carry extra information of weather station design. This method is described in later sections in this chapter. But ICI has the special significance it represents the real packet's functions. It is used to carry additional information that is attached with the packet of weather station design that does not exist in the real packet and has the advantage of supporting simulation control.

In order to apply the ICI format first it will be created from "File" menu, and the "ICI format" to be selected. The attribute has to be added as mentioned above.

4.7.8.2 ICI Content

Whenever interface control information content is created the name of ICI format must be given to that attribute name. The ICI contents have four parameters in its dialog box shown in figure 4.7.8.2.1. The description is optional and others (Attribute Name, Type, Default Value) are provided with their appropriate values.

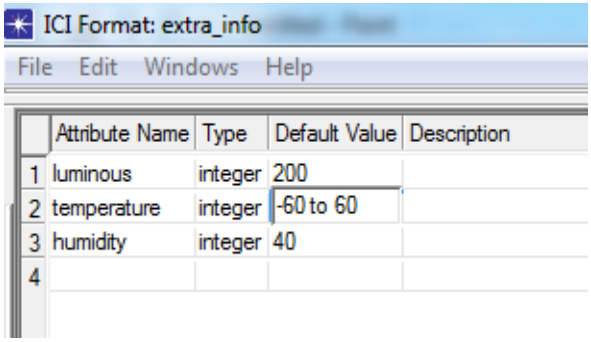


Figure 4.7.8.2.1: ICI Contents

Luminous value is set along with temperature values which set between -60 to 60. For these values of the temperature there is a risk of fire, system understands that for these values the temperature is not normal same is the case with the humidity.

4.7.9 State variables

State variables, unlike temporary variables, retain the information. These are kind of memory forms that permit information to be declared and manipulated directly. Here some static variables are defined because it is the requirement of the code of the project. For example code uses head_dist function and that function requires Distribution* type of variable otherwise the function will not work the same as the case with other functions, for example

ICI functions. Objid is defined which is necessary to determine the object ID because there are parallel streams terminating at the node. Stathandle is defined for the packet count at the receiver side and it also works out the throughput of the specified link. Figure 4.7.9.1 shows "weather station design" state variables.

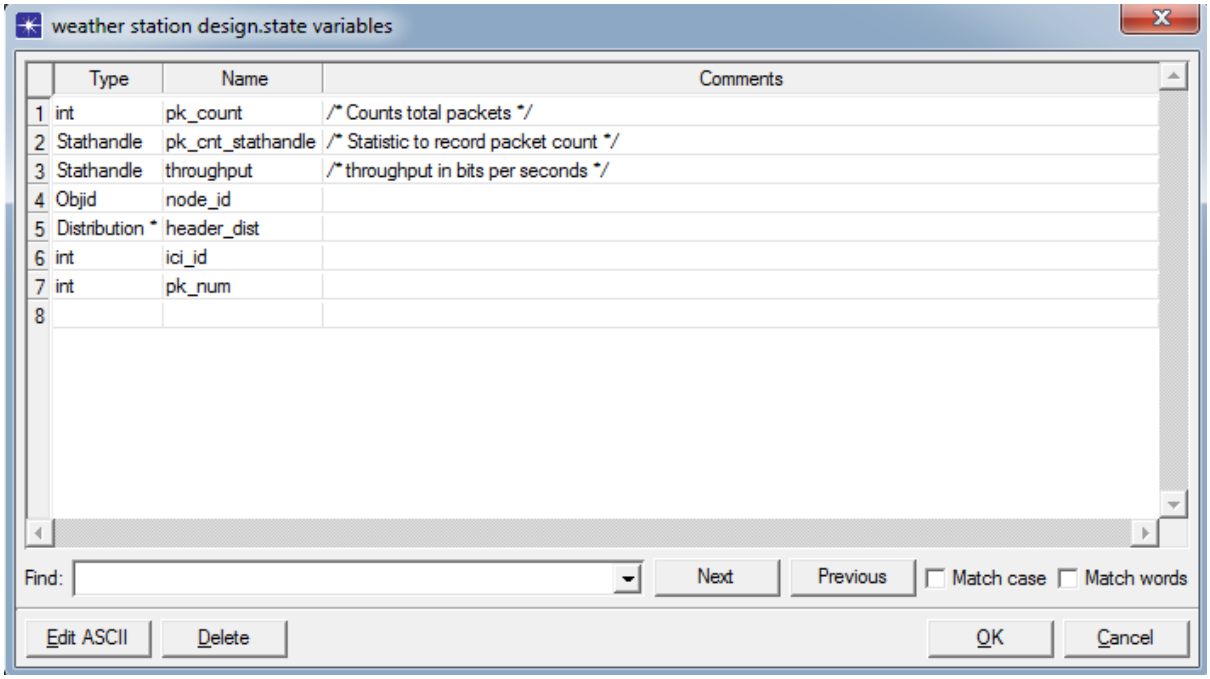


Figure 4.7.9.1: State Variables for the Weather Station Model

4.7.10 Temporary Variables

These variables don't retain information. It is important to note that they are useful for temporary storage. If persistency is needed then these are not useful state variables and the stored information will be lost.

4.7.10.1 Description of temporary variables used in the project code

The project code requires set of temporary variables. This section describes these variables.

According to the code designed, some variables that needed to initialize "OPC_NIL" to "packet *pk" will be returned if control found nothing. Similarly, "header" is declared and one can give any value 1, 2, or 3 as it will not affect the process model, even if it is not declared because the system will give it a default value. Data type Integer with the name "printed" is defined and its values are initialized to zero It is used to check (if it is required) whether the specified object has the printed attribute or not. Packet type wrapper packet, "wrapper_pk" is defined which is initialized to OPC_NIL if nothing returned. Random integer needs to be generated for that purpose "rand_int", "PrgT_Random_Gen" are defined. In order to build the ICI format successfully. It is essential to define ICI data type in the temporary variable

otherwise using ICI function directly in the process model will result into an error message that is “undeclared variable”. Temporary variables code is listed below:

```
packet *pk = OPC_NIL;
declaration of the packet temporary variable
packet *wrapper_pk = OPC_NIL;
Packet type wrapper packet, “wrapper_pk” is defined which is initialized to OPC_NIL if nothing returned.
int printed = 0;
Data type Integer with the name “printed” is defined and its values are initialized to zero It is used to check (if it is required) whether the specified object has the printed attribute or not.
int header = 0;
Data type int with the name header is initialized to zero.
int new_seed = 0, rand_int;
PrgT_Random_Gen    *my_rng;
Random integer needs to be generated for that purpose “rand_int”, “PrgT_Random_Gen” are defined.
Ici  *ici = OPC_NIL;
It is essential to define ICI data type in the temporary variable otherwise using ICI function directly in the process model will result into an error message that is “undeclared variable”.

int temp;
declaration of tem temporary variable
```

4.7.11 Code in the Header Block

The input stream index for lower stream is set to “0”. Note that, the Lower_IN is input stream, it is the logical concept. The type of interrupt is determined by the macro condition. Later condition is checked to what kind of interrupt takes place. However the stream index can be identified in the Node Editor by checking the packet stream’s source stream or destination stream attribute depending on whether the stream is input or output relative to the studied module; in this case the studied module is the P_1 shown above in the Node model.

4.7.11.1 Macro conditions in the header

For the P_1, lower stream index is defined as there is no upper stream index therefore as conventionally has given the value that is index 0. The index should be similar for stream coming in and going out that is why lower_in and lower_out has index 0 in order to make the system understandable. In the next step macros are defined in order to check what kind of interrupt takes place as in the Opnet communication is possible only with the help of interrupts therefore it is necessary to identify those interrupts that's why the macros are defined in the header block commonly.

```
#define LOWER_IN_STRM_INDEX 0
Index 0 is defined for the stream coming in
#define LOWER_OUT_STRM_INDEX 0
Index 0 is defined for the stream going out
#define LOWER_STRM (\
Macro for the lower stream index
  (op_intrpt_type() == OPC_INTRPT_STRM)&& \
     (op_intrpt_strm() == LOWER_IN_STRM_INDEX))
 Macros are defined in order to check what kind of interrupt takes place
#define ARRIVAL (op_intrpt_type () == OPC_INTRPT_STRM)
To check whether the interrupt is a stream type interrupt or not.
Op_intrpt_type ()
```

Purpose

This function, **Op_intrpt_type (),** determines the type of interrupt that currently take place.

The return value is the current type of interrupt.

4.7.12 Code in the receiver state

The following is the code that one needs to put in the receiver state in order to get the special packet. Header will be decoded here. ICI will be extracted from the packet and packet will be decapsulated giving the complete information that weather condition are not normal. This special packet is printed in the simulation console and finally this packet will be destroyed.

4.7.12.1 Explanation of the code given below

The first line of the code is used to get the packet coming from the sender side. In the second line of the code ICI object is obtained which is associated with the wrapper packet. The next line of the code gets the value of 32 bit integer attribute in the specified ICI. In the next stage the value “id” attribute in the simulation console for this ICI object is printed. All signalling information associated with the packet is printed at this stage and after ICI contents are destroyed after they have been extracted from the packet. In the next line code gets the fields of the formatted packet that is received. The code is using if condition and it depends on the value of header. If the header field value is greater than thirty nine it will show the condition of humidity or condition for humidity is true and then print the size of the packet. In the next line condition will be only checked if the first condition comes to be false and if the header value is equal to -60 to 60 that is the condition for the temperature then again. After that print the packet and packet will be destroyed.

wrapper_pk = op_pk_get(LOWER_IN_STRM_INDEX);

Get the wrapper packet which is arrived at the input stream.

ici = op_pk_ici_get(wrapper_pk);

It is used to get the ICI object associated with wrapper packet.

op_ici_attr_get_int32(ici, "id", &temp);

This function gets the value of 32 bit integer attribute in the specified ICI.

sprintf(msg, "ICI id: %d", temp);

Print out the value “id” attribute in the simulation console for this ICI object.

op_sim_message("ICI test", msg);

This finction is used to print all the signalling information.

op_ici_destroy(ici);

Destroy this object afterwards (logically they cannot be destroyed)

op_pk_nfd_get(wrapper_pk, "header", &header);

ICI contents have been extracted so it can be discarded obtains the value of the header field in the specified packet.

op_pk_nfd_get(wrapper_pk, "payload", &pk);

The code gets the fields of the formatted packet that is received.

if(header > 39)

If the header field value is greater than thirty nine it will show the condition of humidity or condition for humidity is true and then print the size of the packet.

print_pk_size(pk);

else if(header ==- 30)

if the header value is equal to -60 to 60 that is the condition for the temperature then again.

print_pk_size(pk);

print this packet which is special packet.

op_pk_destroy(pk);

Finally destroy the packet.

op_pk_destroy(wrapper_pk);

Destroy the wrapper packet.

++pk_count;

Increment the packet count.

op_pk_destroy (op_pk_get (op_intrpt_strm ()));

Determine the packet stream for current interrupt getting the packet pointer and finally destroys it.

op_stat_write (pk_cnt_stathandle, pk_count);

Write out the value of the statistics that is to record the packet count.

4.7.13 WIMAX SUBSCRIBER STATIONS SIDE

This section consists of a number of subscriber stations that are within the access of WIMAX base station who is sending the interrupt message to the subscriber stations. A programme is needed in order to receive the interrupt at the receiver side. For this purpose special KPs are used in order to receive the quick alert from the base station. These KPs are discussed in detail one by one below.

4.7.13.1 Receiver State Code Explanation

A receiver that has used in the receiver state of the process model is described below. Each KP is described with reference to the code of the project extended with the benefits of KP and its purpose, syntax, where it can be used and when it can be used.

Op_pk_get()

Purpose

The packets are arrived and they queued up in manner as they are arrived. This function gets the packet actually it searched the pointer to packet arrived on the input stream. This function return type value is Packet*.

Syntax

op_pk_get (instrm_index)

Explanation and how this KP work in weather station design

The interrupt will not be invoked for the module concerned by simply sending the Packets to an input stream. The bulk of packets sent to an input stream make a queue in that particular order as they arrived. Therefore op_pk_get() gets the packets that is the special packet having the ICI associated with it in order to convey the weather station information following the scheme first come serve first in the input stream; the packets that came late and make up queue at the input stream are achieved by using this KP again that is the successive calls to this KP.

Where this KP can be used!

There are some conditions of using this KP. First of all it is the correct place. Here, the process model must be called on within or a function that has been directly or indirectly called by a process model. However this KP has the facility of obtaining both unformatted and formatted packets.

When this KP can be used!

When an input stream index gets beyond the index of the highest existing input stream, then the new input stream is distributed by this Simulation Kernel. At this situation this KP gives assistance for the reception of the packets whether they are formatted or unformatted on the given number of or unlimited input streams with the help of mechanism called packet delivery mechanism.

Op_pk_ici_get()

Purpose

This KP is used to get the ICI pointer that is already attached with the packet using another KP known as Op_pk_ici_set().

Syntax

Op_ok_ici_get (pkptr)

Explanation and how this KP work in weather station design

It is important to note that ICI information cannot be lost when the packets will be destroyed therefore the ICI information can be used again with other packets therefore the system will keep the template of ICI information that is weather station information to inform user again about the threats caused by weather. System will inform user that there are chances of weather threats again. At the sender side the value can be assigned to ICI pointer only with the help of KP *op_pk_ici_set()*.

Where this KP can be used!

This KP is usually invoked in the process context and therefore used inside the process model. Inside the code where it is essential to get the ICI pointer this KP will be used.

When this KP can be used!

As interface control information carries extra information and their ICI content are associated with the packet therefore it is well to use it with op_pk_ici_set that creates a connection between interface control information that is the weather station design information with the packet. It is so because that weather station design information will be readily available when it is required by packet handling logic.

Op_ici_attr_get_int32()

Purpose

It may be the int32 or int64 depending upon what is defined in that particular ICI. The purpose of this KP is that it is used to get the value of a 32-bit integer attribute in the predefined ICI.

Syntax

op_ici_attr_get_int32 (iciptr, attr_name, value_ptr)

Explanation and how this KP work in weather station design

As the ICI content information contains the attribute name and its value therefore he specified attribute name must be one of the attributes defined in the ICI format for example temperature, humidity and luminous in the case of this project.

This KP is particular useful for the weather station design because it conveys the latest information. For example, if a process creates an ICI, assigns its attributes, installs it, sends out a packet, and then modifies an attribute of the ICI before the packet is received, the modified value will be the one provided when the process model receiving the packet accesses the ICI via this KP.

Where this KP can be used!

This KP is used inside the process model more specifically one can say that it is used at the receiver side of ICI based communication. This KP is not restricted to single way communication as weather station information needs to broadcast therefore this KP provides the mechanism to share the weather station information in a multi communication way or it can be used to broadcast information as ICI can be shared among the multi processes. Therefore this KP is suitable when a process gets interrupt or weather station information from the specified ICI and also returns information by setting attributes in the same ICI.

When this KP can be used!

It is used when it is needed to determine the values of the ICI's 32-bit integer attributes for a process when it has received interrupt. As this KP return value can called immediately within no time therefore it is the best option to use it for the weather station design in order to convey information immediately that weather is abnormal.

Op_sim_message()

Purpose

This KP simply prints the message and show it on a message console or the Simulation Progress dialog box.

Syntax

Op_sim_message (line0,line1)

Explanation and how this KP work in weather station design

Here in the weather station design it this KP is used to print all the signaling information that is associated with the packet.

Where this KP can be used!

It is used anywhere in the process model. As in this weather station model it is used in the receiver side to print out the receive message and show it in the console.

When this KP can be used!

. It is the best option to use when it is needed to print the any two line of the text message. This KP is an alternative approach for printf() command because it handles well and sufficiently any Project Editor situation.

Op_ici_destroy()

Purpose

This KP destroys the ICI content so that the new ICI can be created again and can be reused. For this purpose one can say that it Deallocates the specified ICI, which was once allocated releasing associated memory resources for other purposes which once was occupied.

Syntax

Op_ici_destroy (iciptr)

Explanation and how these KP works in weather station design

It is necessary to destroy the ICI information otherwise outstanding ICI will be created which will stop the simulation when running again and the system will not accept the allocation requests for the new objects therefore it is essential to release memory. In this weather station this information will be destroyed after printing in the console. In reality ICI information cannot be

destroyed but as mentioned can be shared among the multi processes which is the case of weather station design.

Where this KP is used!

This KP is mostly used at the receiver side but it is still used in the process model rather than any diagnostic or header block. It is used in situations where memory recycling is essential to make a place for the new objects or to allocate memory to them as it was created initially for the ICI content (which is the weather station information).

When this KP is used!

When it becomes essential to allocate the memory to the other objects for the next simulation then it is necessary to release the memory then this KP will be used. Op_pk_destroy() is not alternative approach to destroy ICI therefore it is essential to use this KP op_ici_destroy() wherever ICI format is used. This KP must be called whenever there is a need to get rid of ICI. After invocation of this KP, the specified ICI pointer should be considered invalid.

Op_pk_nfd_get()

Purpose

This KP gets the value from the header of the packet which is already set at the sender side.

Syntax

op_pk_nfd_get (pkptr, fd_name, value_ptr)

Explanation and how this KP works in the weather station design

This KP supports value access for all five types of packet fields. It is necessary to extract the value from the header of the packet before applying the condition. Therefor this KP is used to get the value from the header of the packet to check what the temperature, humidity value is. This KP has the wide range of facility of supporting different types of data which includes integer, double, packet, information, and structure.

Where this KP will be used!

It is used where it is essential to get the value of packet field and it is of supreme importance for the purpose of carrying data.

When this KP will be used!

This KP is mostly used on such situation when the packet is at the stage of processing, storing and requires acknowledgement message. Therefore the best option to invoke this KP is when the packet will arrive at the input stream of queue and processor.

Op_pk_destroy ()

Purpose

The purpose of this KP is to release the memory to make a space for the new objects.

Syntax

Op_pk_destroy (pkptr)

Explanation and how this KP works in KP design

The packet is destroyed if it is not assessed. It is not sent to module, and its place is removed from memory. In other words one can say that memory is released this is the data information here it is the packet which is destroyed at the receiver side it is the packet that is attached with the ICI contents. As mentioned earlier memory will be released again to make a space for the new objects so that the memory will be allocated. Routes associated with the original packet are also destroyed. It is important to note ICI contents cannot be destroyed even with the KP op_ici_destroy that is the weather station information associated with the packet or one can say that interface control information contents.

Same is the situation with this KP as with the op_ici_destroy that it is essential to destroy the packet otherwise it will lead to the creation of the outstanding packets that will stop the progressive simulation which further results in such situation that memory will not be allocated to new objects.

Where this KP can be used!

In this project this KP is used in the receiver state to give assistance to get the memory recycled for the new packets because the old packets are of no use.

When this KP can be used!

This KP should be invoked after a packet has been accessed for the last time. As soon as this KP returns, the specified packet pointer should be considered invalid in the context that made the call.

4.7.14 Code in the Sender State

The following is the code that one needs to put in the sender state to send the quick notice at the receiver side. Header value will be set here. ICI will be associated with the packet and the packet will be encapsulated and finally encapsulated packet will be sent to the output stream that will leads to the input stream of destination module (receiver).

4.7.14.1 Explanation of code in the Sender State

The first line of code is used to record the statistics. The second line of code explains that the code gets the node object ID and stores it in the node_id state variable. Distribution handle variable uses the distribution package; here from the distribution package specific function op_dist_load is used. This functions load random integers between zero and forty randomly, and condition for these numbers is checked on the receiver side after reading the header value. After that interval for the numbers that are generated randomly to check for weather station condition is assigned. In the next step of code number is destroyed. In the next steps fields of the packet will be assigned value. In the same the next lines of code describes the creation of ICI setting of value and associating with the packet. Finally the packet will be sent out. The last lines of code describe the conditions of weather stations and three cases. If any condition from these three cases comes to be true then the message will be printed in the console.

```
throughput = op_stat_reg ("Throughput (bits/sec)",\
        OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
node_id = op_topo_parent (op_id_self());
header_dist = op_dist_load("unifrom_int",0, 40);
/* create a new random number generator */
my_rng = op_prg_random_gen_create (new_seed);
/* generate a random integer in the interval [1,6] */
rand_int = (op_prg_random_integer_gen (my_rng) % 6) + 1;
/* destroy the random number generator */
op_prg_random_gen_destroy (my_rng);
op_pk_nfd_set(wrapper_pk, "payload", pk);
ici = op_ici_create ("extra_info");
op_ici_attr_set_int32(ici, "id", ++ici_id);
op_pk_ici_set(wrapper_pk, ici);
op_pk_send(wrapper_pk, LOWER_OUT_STRM_INDEX);
pk_count = 0;
pk_cnt_stathandle = op_stat_reg ("packet count",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
```

```
printf ("Weather station design (%d)\n");

printf ("\n");

printf ("Amir alarm pk\n");

printf (" case1luminous\n");

printf ("case2 temperature value is not normal 30\n");

printf ("case3 Humidity is increased upto 40%\n");

printf ("Chance of fluctuation with these values in next hour again \n");

printf ("United Kingdom\n");

{

printf ("\n");

printf ("\n");

}

{

int a=1 ;

if (a < pk_count)

        {

        printf ("<<< Quick Notice::temperature and Humidity is not normal
>>>\n");

        printf ("<<< Certainity::likely to happen >>>\n");

        printf ("<<< Urgency::Immediately >>>\n");

        }

else
```

```
        {
         printf ("<<< Quick Notice::Luminous  >>>\n");
        printf ("<<< Severity is high estimated millions of volts may produce
>>>\n");
        printf ("<<< Area Description::circle >>>\n");
        }
}
```

4.7.14.1 WIMAX ELECTRONIC SENSOR AND WIMAX BASE STATION SIDE

ICI is used to complete this interface. Sensor is attached to the WIMAX base station. The packet coming from the sensor to the WIMAX has the interface control information associated with it. Sensor will sense the weather abnormal condition and send the packet and ICI as a whole to WIMAX base station which will broadcast this packet along with ICI content to the subscriber stations within its access.

4.7.14.2 Sender state Code Explanation

Op_dist_load()

Distribution handle variable uses the distribution package; here from the distribution package specific function op_dist_load is used. This functions load random integers between zero and forty randomly, and condition for these numbers is checked on the receiver side after reading the header value. The code gets the node object ID and stores it in the node_id state variable. Code also loads a uniform distribution between integer zero and forty.[5]

Op_pk_nfd_set()

It gives assistance to set the value of a field within a packet. For the formatted packet, only the value of field needs to be specified. But if unformatted packet is used, the type and size of the field is essential to describe.

Purpose

This KP is used to assign the value to the field of the packet. It can be any value for example integer double here in this weather station design integer value is used to for the temperature value.

Syntax

op_pk_nfd_set (pkptr, fd_name, value)

Explanation how this KP work in weather station design

However this KP has the adjustment of setting value at run time. This KP is used in the weather station design because the logic that this KP possess adjusts the value assignment based on the header or the internal format of the packet and hence this KP is the best option to use for weather station design as it is able to adjust the weather station parameters according to the situation at run time and one can say that this KP works even the values of packet field are not predefined.

Where this KP can be used!

This KP is used at the sender side inside the process model but the important thing is this KP is used often for the formatted packets. Here in this project it is used in the sender state It can be used for the unformatted packet but the two things need to be specified that is type and size of the field.

When this KP can be used!

This KP is used when it is required to set the value of fields of the packet and hence this KP is needed and it provides the mechanism of adjusting value. When fields are assign value it means they are encapsulated and after that this KP works on the encapsulated packets and invoking this KP releases the ownership of the encapsulated packet and one cannot use these packet before DE capsulation.

Op_ici_create()

Purpose

This KP is used to build interface control information by the specified ICI format. It builds new ICI with already described structure of ICI format.

Syntax

op_ici_create (fmt_name)

fmt name here in this project is the extra_info that is described in the code.

Explanation and how this KP work in weather station design

It is used in sensor side of this project and has the vital role of building ICI inter process communication. It manages the interrupts if parameters process model of ICIs changes and in this way in builds up the ICI communication by managing additional interrupts.

Where this KP is used!

This KP is used to give assistance to neighbouring processes by using new ICIs communication. It is used where it is needed manage the process models ICIs by using some interrupts. This KP is again not used in the header or function block of the process model but used within the states.

When this KP is used!

In the process model it is required to schedule interrupt in this communication. This KP should be invoked to manage that newly build ICI communication. It is important to note new

ICI is allocated using this KP and as its return type is the pointer to newly created ICI its pointer allows the assignment of predefined ICI attributed in the process, therefore it is invoked when the values are set by the KP op_ici_attr_set_int32.

Op_ici_attr_set_int32()

Purpose

In the specified ICI it is needed to allocate value and this value is 32 bit integer, hence this KP is used to do this task.

Syntax

op_ici_attr_set_int32 (iciptr, attr_name, value)

In this project the syntax is (ici, “id”, ++ici_id);

Explanation and how this KP work in weather station design

Using this KP assigns the value and attribute name to the specified ICI. In other words this KP laid down the foundation of the interface control information contents that are attached with the packet and as a whole making the packet for weather station design. The important thing to note is specified attribute name must be one of the attributes defined in the ICI format. As in the weather station fluctuation in the temperature and humidity value may occur therefore this KP will overwrite to he previously assigned value to the attribute and hence system will work perfectly at run time. This new value is approachable to other processes but the condition is that the process should keep a pointer to the interface control information.

Where this KP is used!

This KP forms the sender side or transmits side of ICI based communication and hence used at the transmitter side .As it is mentioned above this KP can overwrite therefore the modified ICI might still be pointed to by its originating process and hence this KP can be used to set results that the originating process can then obtain through the KP *op_ici_attr_get_int32()* which will be used at the subscriber side. In this way this KP has the capability of allowing bi directional communication to occur in this ICI based communication with a forced interrupt, and this makes results achievement quickly which is of supreme importance in this project.

When this KP is used!

It is not only used in when ICI based communication but also used in other communication when it is essential for a process to set the contents of parameters required for particular design.

Op_pk_ici_set()

Purpose

This KP laid down the foundation of the project because this KP builds a connection between ICI contents and packet so that one can have the combined effect of both packet and interface control information as a whole during simulation.

Syntax

op_pk_ici_set (pkptr, iciptr)

In this project its syntax is op_pk_ici_set (wrapper_pk, ici)

Explanation and how this KP work in weather station design

By default the OPC_NIL is the initial value for the ICI pointer in a newly created packet is OPC_NIL. ICI pointers they play important role in this project design as they cannot be destroyed once they are connected with the packet with the help of this KP. If one make another call of this KP, ICI pointer still remain in the packet and this make weather station to have template inside and invoke this value again when again temperature, luminous or humidity become abnormal. Therefore this KP is the best choice in this project design.

However at the receiver side it is generally done that ICI is destroyed which means ICI is not legal valid in association with packet.

Where this KP is used!

It is used in ICI based communication where it gives assistance build a connection between a packet and an ICI and finally has combined effect during simulation. This KP is again used in the process context but it forms the transmit side of ICI based communication.

When this KP is used!

This KP is the best choice when a packet and an ICI arrive with the same interrupt, but the packet must wait to be serviced. Before the packet is queued, the ICI pointer can be set in the packet and retrieved with the KP *op_pk_ici_get()* when service is to begin.

Op_pk_send()

There are three main tasks that this KP performs. These tasks are given below

1) It forwards the packet using output stream to the input stream of destination module.
2) At the destination side it manages and controls the arrival of the packet.

3) Finally it loses the right to possess the packet that means it releases the ownership of the packet when the process will be invoked.

Purpose

This KP is used to send the packet by passing through the output packet stream and it manages the packet arrival at a destination module for the current simulation time, and releases ownership of the packet by the invoking process.

Syntax

op_pk_send (pkptr, outstrm_index)

Here it is Lower_OUT_STRM_INDEX

Explanation and how this KP work in weather station design

When packet comes to this KP cease the right of possessing the packet and it happens when this KP is invoked and after that the right of possessing the packet is given to the destination module which in this project is subscriber side of weather station. It is the final stage at the sender side after invoking this KP means encapsulated packet is now delivered at receiver side where the packet is decapsulated read out printed and destroyed.

Where this KP is used!

This gives assistance to packets for sending then to their destination that are passing within the node among different modules and that are linked by packet streams. However, KP gives assistance to packets to be forwarded to other processes that are under execution in processors or queues, or to transmitter modules. This KP requires a process context, so it must only be invoked from a process model, or a function that has been invoked by a process model. Both formatted and unformatted packets are supported by this KP.

When this KP is used!

This KP is the fundamental means of packet forwarding used in simulations. Here the destination is the lower out strm index and is defined in the header block. In many cases it is used at the end of code when it is needed to schedules the specified packet's arrival at the destination module.[5]

4.7.15 Design Methodology2

As mentioned above one can use header field to carry information, this method can be implemented by building and accessing the formatted packet in the process model along with the method to create and apply function in the process model. In source state values are set into the header. Set function is used for this purpose. In receiver state value is

obtained and get function is used to get the value that was set inside the header at the source side.

4.7.15.1 Code in the Function Block

In this approach one use the header field to carry the information with the packet. It models the real packet's functions. If one is following the design methodology2 then in function block it is needed to add code to implement the function. This method is not described in the detail as other approach is used for weather station design.

4.7.15.2 Explanation of the code given below

The FIN and FOUT are used here to enable the Opnet debugging kernel to print out function information. The code describes if the value of the header field is greater than five, and then prints the size of the payload packet. Similarly one can set any type of value for the temperature and value can be printed for that quick notice alert saying that weather conditions are not normal.

```
static void print_pk_size (packet *pk)
Static keyword is used to save the different process models as different source file.There is no need to return value here that's why void is used because the purpose is to print the message.
        {
          char msg[128];
Defining the length of the message, it can be more or less but not fixed.
          FIN(print_pk_size(pk));
"FIN" is used in OPNETfunctions to enable the OPNET debugging kernel.
to print out function information.
          sprintf(msg, "payload packet size: %d", \
                    op_pk_total_size_get (pk));
Print the packet whose name is payload packet.
          op_sim_message ("Wrapper packet header > 5", msg);
Print the packet if the header value is greater than five.
          FOUT;
The macros "FOUT" is used in OPNET functions to enable the OPNET debugging kernel to print out function information.
}
```

4.8 Chapter Summary

This chapter has started with aims and objectives of the Weather Station (WeS). A novel methodology approached has been adapted to model WeS. WeS model has been described fully with reference to WiMAX definition in section 4.3. The implementation has been fully presented in section 4.6. The design of process model is extended with the code explanation in the sender side and the receiving side and handling process is described. The code presented in section 4.7 with programing frame of work. The model has been programed using Opnet Proto-C and compiled successfully on Opnet environment.

Chapter five: Results and Discussion

WeS model has been designed and simulated successfully based on program technicalities and functionalities issues related to WiMAX protocol requirements and weather forecast data.

5.1 Results Overview

The model has been compiled under Opnet environment version 17.4. All syntax errors have been dealt with based on Proto C format. Currently, there are no syntax errors and results appeared successfully.

The total duration of the simulation is run for 100 seconds which is considered long enough as the number of subscribers in this model is relatively small.

WeS model has been coded to indicate that certain values are special for which the network consider those values as abnormal and report to the subscriber stations in this weather station design. For example temperature values between -60 to 60 are not normal and for humidity 40 or above are extreme situation.

5.2 Results of Three Cases

There can be three cases, all are described below:

1) Luminous threats: Section 5.2.1 presents the results for this case with print out of the results
2) Temperature threats: Section 5.2.2 discusses three temperature values, i- -45, ii- 55 & iii- -30
3) Humidity threats: Section 5.2.3 presents the test for WeS for the humidity condition.

As all these three cases are set in the contents of interface control information, therefore after running the simulation, results are shown in figure 5.2.1. The message is printed in the console without error which shows the simulation is compiled successfully.

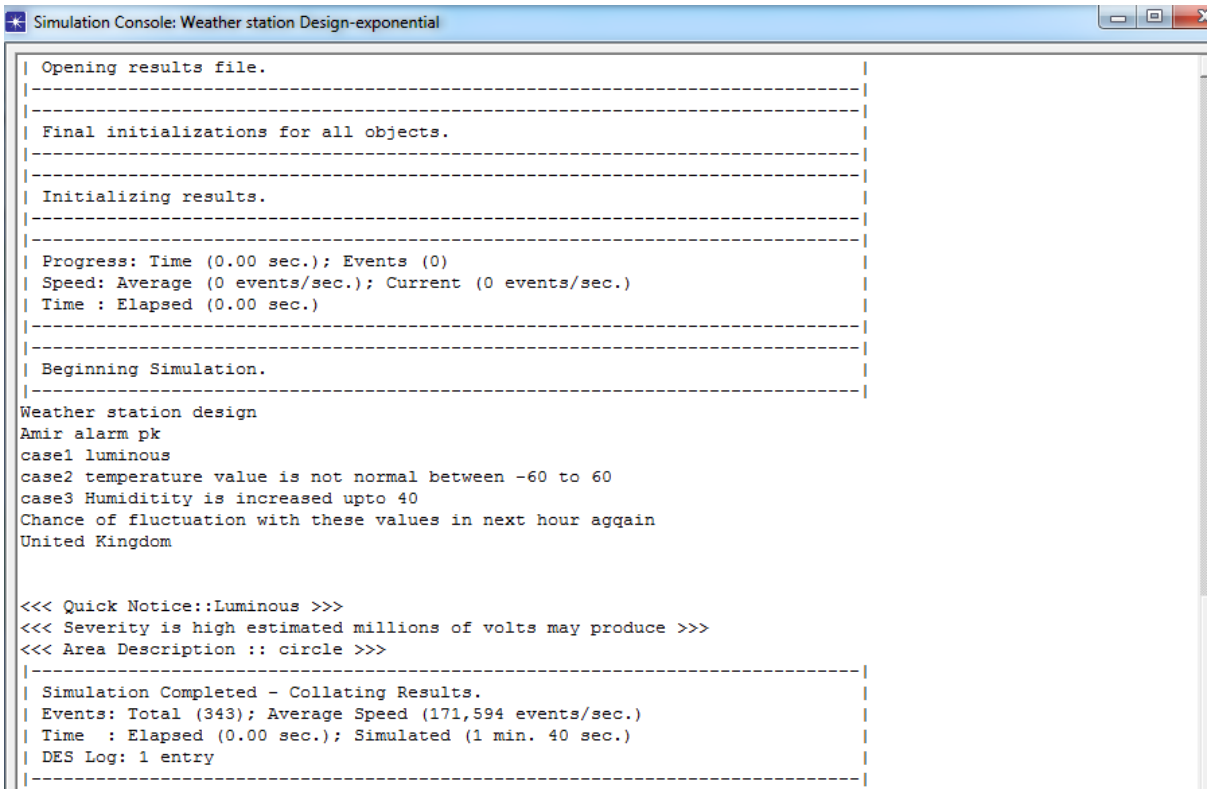


Figure 5.2.1: Case Example

5.2.1 Case1 Luminous

The Weather Station has received information that passed to WiMAX base Station. The information has been compared with a built in table of data relative to different extreme cases. This particular message that printed in the console clearly indicates that there is a threat of luminous that can destroy infrastructure. The message also explains the shape of the area where threat has happened. As the Opnet communication is based on the event results hence in the simulation dialog box total events are shown that are 343. The dialog box in figure 5.2.1.1 also shows the speed in events per second.

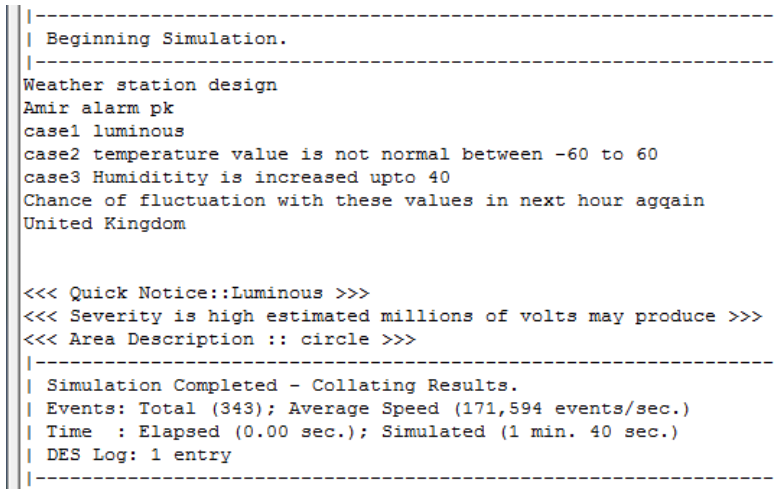


Figure 5.2.2: Case1 Luminous

5.2.2 Case2 Temperature

Simulation is run for different values of temperature from negative to positive. There are some functions in the process model that might overwrite the ICI content and create new ICI or assigns new value by using some other functions. Because there are no strict ownership controls on ICIs, as there are on packets, any process that has access to an ICI's address can do these operations. For example, an ICI may be destroyed by the process that created it, or by one of the processes that have received it. Similarly, multiple processes can concurrently have access to the same ICI and cause modifications that will be visible to each other. In order to avoid such situations system checks the value increment the encapsulated packet also at the receiver side packet is DE capsulated and ICI contents are detached.

All it happens at run time during simulation. This time at run time the network detects that the temperature value is not normal that is -45 ^{o}C and there is a chance of fire.

Figure 5.2.2.1 shows the console with temperature value of -45 ^{o}C.

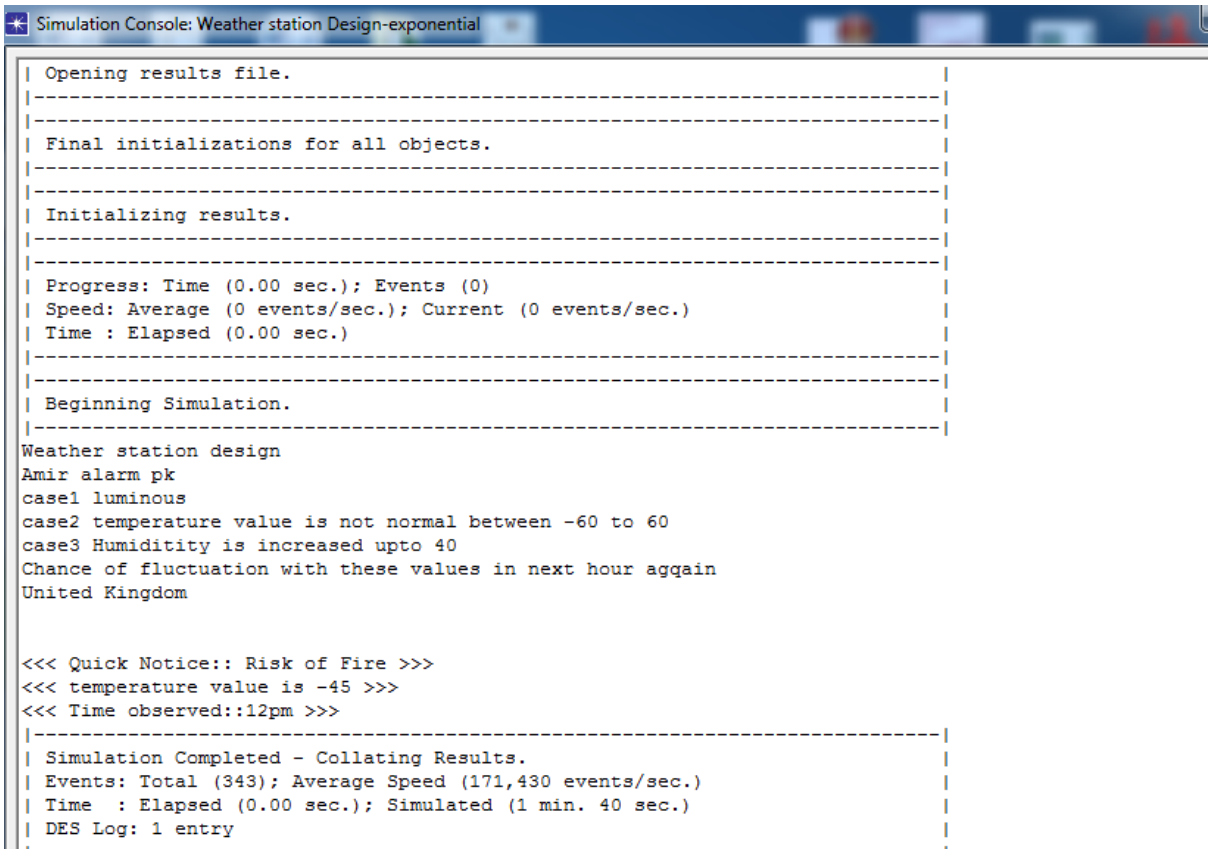


Figure 5.2.2.1: Case2 Temperature 1

In figure 5.2.2.2shows, the time simulation dialog box shows different temperature value which is 55 ^{o}C. First network check which case is true, as in the following case the temperature condition is true it further checks the integer value which value matches the ICI content. In fact ICI controls the information between processes. They are used for the interprocess communication specific for conveying information which controls layered protocol interfaces. ICI formats are referenced in calls to KP from within process models. These ICI are associated with the event. They associate additional parameter with the packet transfer. By adding the additional parameter along with the packet ICI also enhances the data transfer that's why they are very efficient for working with the risk management or emergency conditions. These ICIs are not event specific; they can be used in any case and it is finally shown in the dialog box. Here in the following case 55 ^{o}C value matches with ICI content which is previously created or newly created during run time.

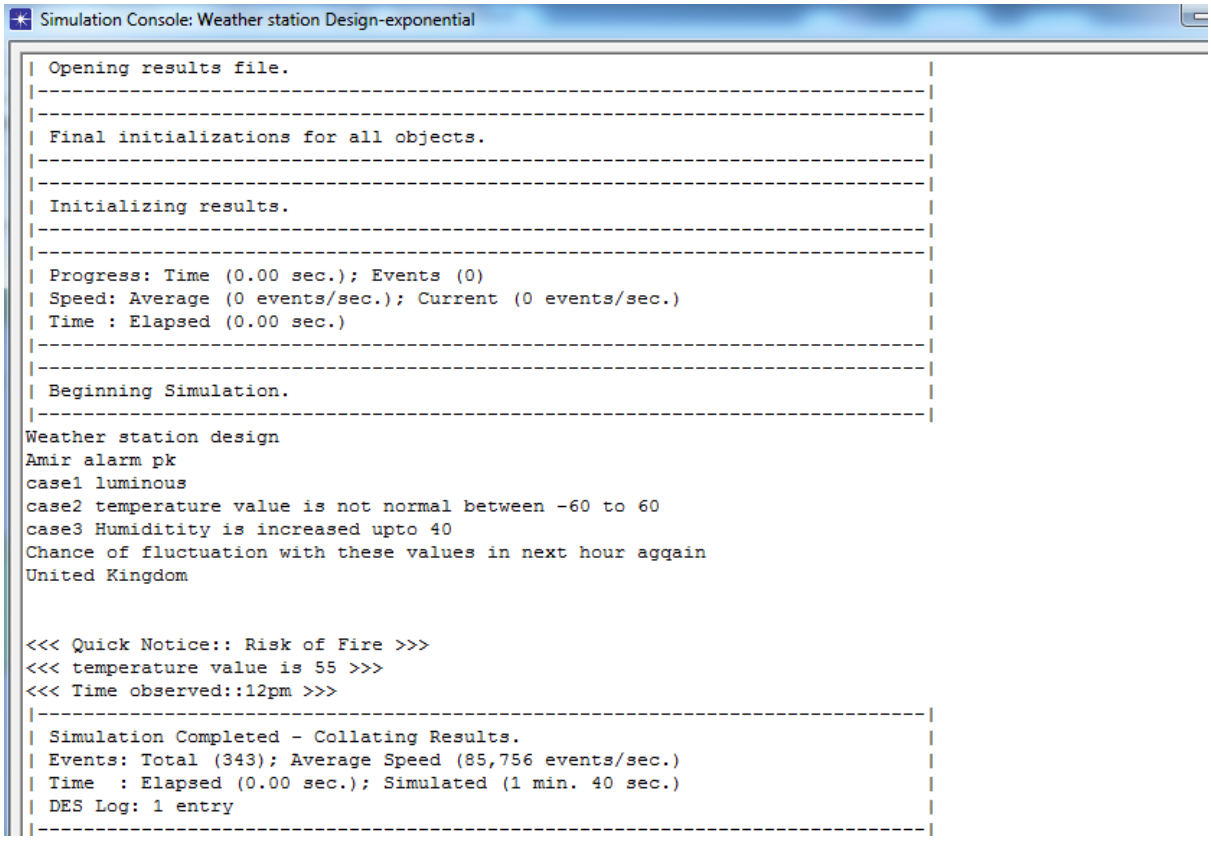


Figure 5.2.2.2: Case2 Temperature 2

The simulation has been run again but in this case the condition of temperature comes out to be true. Therefore the message printed in the console is shown in figure 5.2.2.3. The dialog clearly indicates that there is a risk of fire as the temperature value is -30 ^{o}C and the time when this happened is also described in this simulation box.

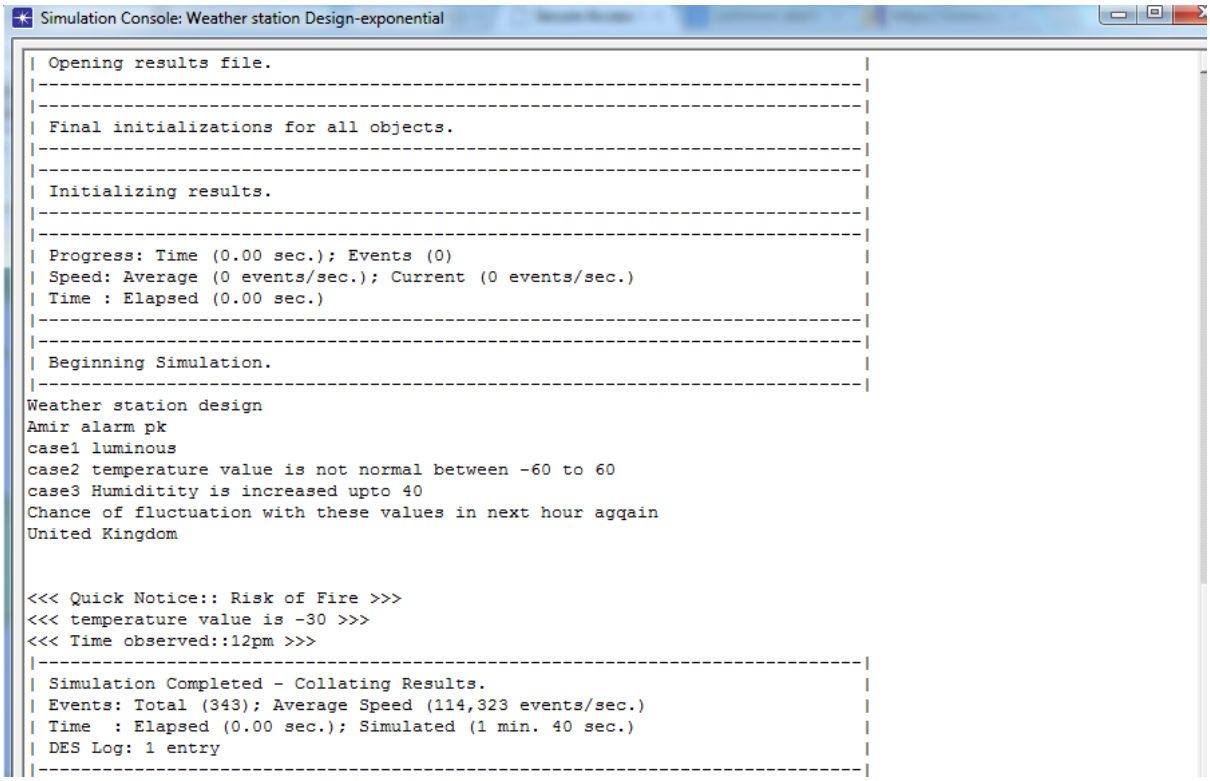


Figure 5,2.2.3: Case2 Temperature 3

5.2.3 Case3 Humidity

In order to check that the systems work properly the systems is examined for the humidity condition. Network at the sensor side checks conditions for the temperature, luminous but this time the condition of humidity comes to be true and the message is displayed in the console as shown in figure 5.2.3.1.

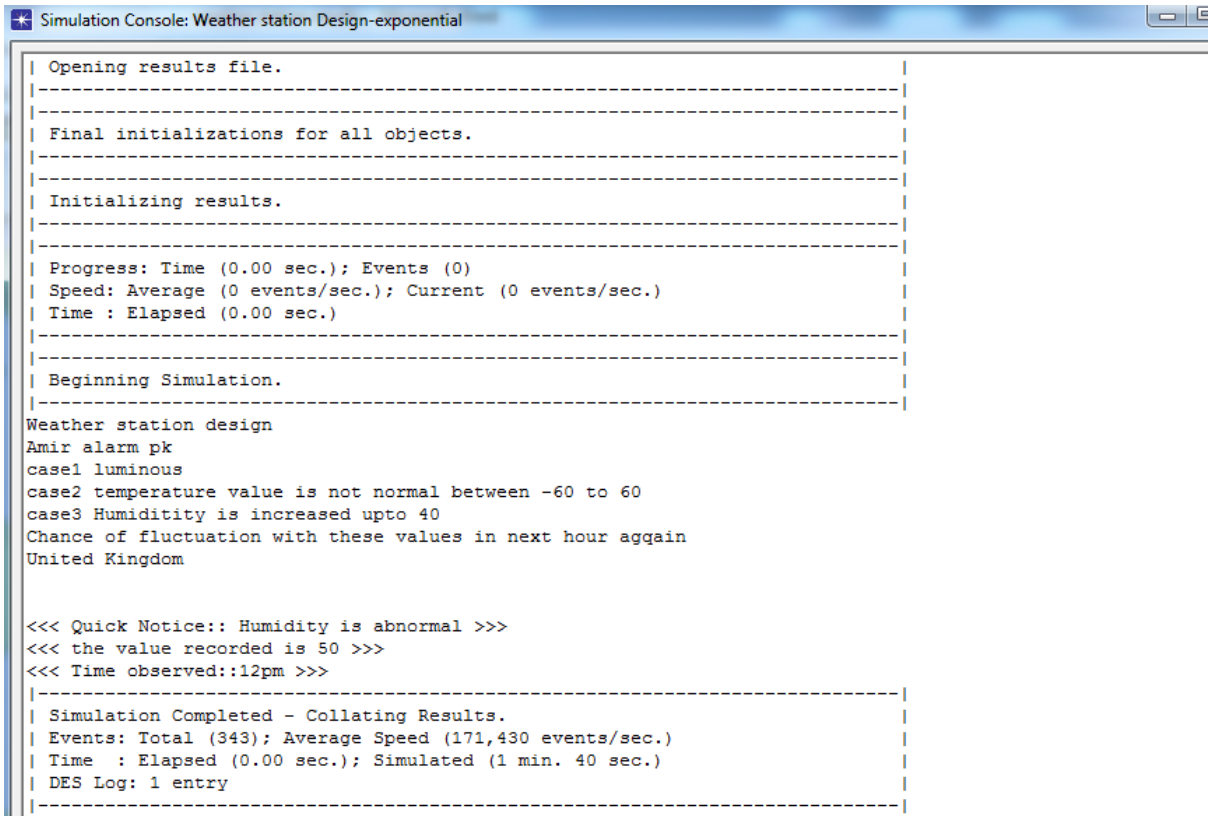


Figure 5.2.3.1: Case3 Humidity

5.3 Discussion

The results that have been presented in sections 5.2.1, 5.2.2 and 5.2.3 proves that WeS model is working according to plan set in chapter 4. As WiMAX protocol is rigours and approved by IEEE and labelled as IEEE 802.16e, WeS is a complimentary successful addition to the same protocol.

However, the above approach used to accomplish this project is limited for certain operations because it does not analyse the network for some statistics. Many types of reports should be included in network design for example the name of these reports are given below

1) BS admission control statistics
2) BS admitted connections
3) BS rejected connections
4) MS power consumption report
5) MS power saving report

Including above statistics and run simulation one can have the high level summary of accepted and rejected connections in the network and using this approach one can determine which subscriber station did not receive the alert from the base station. This will

act like TCP protocol and it will be used for reliable communication. High level summary of network will explain the uplink sub frame capacity and downlink sub frame capacity. The three sector base station will generate report for each sector. The power saving report will give the total time spent in each power saving state by the selected MS. In this way the power consumption report gives the total transmission and the operating power consumption of the selected MS where transmission consumption is based on the actual transmission power used by the MS and the operating power consumption is based on the SS operational power settings.

Due to model limitations for WIMAX, the following features have not been implemented:

- Network-assisted initial ranging during handover
- Base station-initiated periodic ranging
- It is not possible to perform the frequency division duplex mode

The above results show that the system is working on run time conditions according to the objectives set earlier. Interface control information and use of suitable KP makes the design applicable for real time applications. ICIs are dynamic objects that are created by processes that want to associate information with events that they schedule. Because there are no strict ownership controls on ICIs, as there are on packets, any process that has access to an ICI's address can do these operations. For example, interface control information may be destroyed by the process that created it, or by one of the processes that have received it. Similarly, multiple processes can concurrently have access to the same interface control information and cause modifications that will be visible to each other.

Using some protocols the above systems, WeS can be implemented with MAC layer of WiMAX base station. The WiMAX base station process model has three main processes:

- WIMAX_ MAC
- WIMAX_bs _control
- WIMAX_ss_control

These three main processes are developed and included in Opnet WiMAX model. Hence, there is no value to re-create them and include them to WeS for this research project.

The first one deals with the packet delivery, data forwarding, and band request mechanism.

The second one is the child process of WIMAX mac that implements the WIMAX functionality on the WIMAX base station for example admission control, responding to ranging messages etc.

WIMAX_ss_control is the child process of WIMAX_ mac that implements WIMAX functionality on subscriber stations. The above designed process model can be used with this child process of WIMAX_mac in order to make the design more efficient.

WeS model with WIMAX_ MAC, WIMAX_bs _control and WIMAX_ss_control represents an efficient design that completes the interface of 4G technology by embedding it into WIMAX_bs_control using proper protocols. The system code implementation of WeS brings the implementation a step towards hardware implementation, given the right hardware platform.

5.4 Chapter Summary

This chapter has presented the results of the new model of the Weather Station (WeS).

The model has been compiled and simulated and the results have been resulted accordingly

In order to verify that results obtained are according to the required design, different cases are performed that analysed the design explicitly. The use of the Interface control information package which is completely defined in Opnet 17.0 meets the objectives laid down in section 5.2 and its subsections.

The discussion in section 5.3 clearly conveys the message that choosing WiMAX is more practical for the disaster management as other methods are not feasible for WeS and it shows that WeS is WiMAX compatible and ready to run for 4G applications.

Next chapter gives a conclusion to the thesis in hand along with future work.

Chapter six: Conclusion and Future Work

6.1 Conclusion

The aim of this project is to design a digital signal processing system that extracts weather information directly from the surroundings which are related to extra-tropical cyclones to provide instant warning using WiMAX, IEEE 802.16e network standard.

After thorough survey in chapter 1, section 1.3, it has been revealed that the current network emergency operations do lack fundamental features that are related to the purpose of such systems. The current emergency network operations do not react with the weather information surrounding the station, the service provided limited with locations and not freely available. CloneWAN has provided a successful network model for the case of the UAE.

Hence, the search is still on for better alternatives. Chapter 3 presented a novel approach to WLAN that will clone the current setup in case of emergencies but based on new network set as described in Chapter 4, that takes weather information, react accordingly and send an alert in case of extreme conditions. This new WAN is called CloneWAN. For full realization, the project would have taken the two routes, the hardware implementation and software modelling routes. However, due to the cost, time and delivery limitations, the software modelling has been adopted throughout the project. Due to its flexibility with implementation, wide range of built-in interrupts/functions and ease of displaying results, the software platform chosen for this project is Opnet. Hence, Chapter two focused on describing Opnet at a hierarchical level.

Due to the hierarchical nature of Opnet, top down implementation was adopted. Chapter three presented a new network technology, CloneWAN system. A UAE was used as a case study to apply and study the results of CloneWAN. The operation of CloneWAN was modelled, simulated and thoroughly studied in sections 3.1 and 3.3. The results presented in Chapter three shows that CloneWAN is an attractive alternative to the operations listed in the survey in chapter one, section 1.3. CloneWAN is based on the rigorous WiMAX standard. WiMAX are formed by set of base stations network with set of subscribers on a range of cell sizes. These base stations are not capable of obtaining weather information directly from their locations. A new station has been designed to present the Weather Station (WeS). WeS is capable getting weather information directly from the surroundings via set of sensors. The full hardware design for WeS has been presented in section 3.4

WeS model has been the focus for Chapter four. The heavy emphasis on the Opnet design methodology with detailed description achieved may be the goal of the project. The beginning of the project is supported by the description of WIMAX extended with the concept and mechanism of Opnet. The middle of the project is supported by the various designs. Finally the end section describes the results, analysis and model limitations.

The report describes the choice of technology “WiMAX” based on its benefits and efficiency that is properly fitted to the project title weather station. The layered architecture of technology is discussed in a great detail. Its characteristics are explored with diagrams showing attributes of WiMAX technology using the Opnet software. The software architecture is presented from the basic level to advance. Opnet Modeller provides a comprehensive development environment for modelling and performance-evaluation of communication networks and distributed systems. The package consists of a number of tools, each one focusing on particular aspects of the modelling task. These tools fall into three major categories that correspond to the three phases of modelling and simulation projects model specification, modelling communication with packets, data collection and simulation and analysis. Like all other subsystems in Opnet Modeller models, processes are driven by events. When an event is actually delivered to a process, it is termed an interrupt. The reasons for which a process may be interrupted vary widely from process to process, and most processes themselves expect several types of interrupts signifying different conditions to which they must respond. Therefore, one of the first actions that are generally taken by a process upon being interrupted is to determine what type of interrupt has occurred. A process follows an alternating cycle of invocation and rest periods. Invocations may occur on an arbitrary basis depending on the timing of externally and internally generated events. A process always begins the simulation in a resting mode, waiting to be invoked; a process that is waiting in these conditions is said to be blocked. Invocation allows a process to resume execution and to do new actions. After these actions are completed, the process must again block, returning control to the Simulation Kernel so that other events in the system may be executed. Because these subsequent events may be scheduled for arbitrarily near future times, and even for the same time as the current event itself, the invocation must occur without allowing any time to elapse. The state variables, temporary variables macros in the header are defined separately under headings in order to have their purpose and clear understanding for the reader. Step by step methodology is given for the design of the project that anyone can follow to approach this design. This procedure includes the node model and its mechanism, process model and its mechanism. Nodes are created as instances of node models, meaning that a node model is the blueprint for all of the individual nodes of a particular type. Node models are defined as a collection of modules

representing distinct functional areas of the node. Certain modules are limited in the types of behaviour they can represent; for example, the various transmitters and receivers represent interfaces to links defined in the network domain. In the process model Proto-C language is used to design code. Proto-C provides a flexible platform that has the ability to model a wide range of systems. The language provides specific support by adopting a state-transition approach which is well-suited to discrete event systems, and by supplying a number of Kernel Procedures (KPs) that are oriented toward network and distributed systems modelling. The core of the project is the interface control information with the help of this approach code of the project is designed.

The main object of the project is to generate the interrupt in case of hazard condition that was designed using ICI. The concept of ICI is used in order to accomplish the project and a dedicated package of KPs supports operations on ICIs such as creation, setting and getting of attributes, destruction, installation, etc. This capability allows information to be transferred from the context where an event is generated to the context where it later occurs. The two contexts may be distinct modules within the same node or in separate nodes, or the two contexts same module executing two separate events at different times. Therefore, an ICI can be viewed as playing a role in a communication mechanism between contexts, where an event provides the active notification. ICI is used to complete this interface. Sensor is attached to the WiMAX base station. From network point of view the packet coming from the sensor to the WiMAX has the interface control information associated with it. Sensor will sense the weather abnormal condition and send the packet and ICI as a whole to WiMAX base station which will broadcast this packet along with ICI content to the subscriber stations within its access.

The sender code is designed for the process model and the receiver code both sides (sender and receiver) code functions are explained separately in sections 4.7 and 4.8 to have the better understanding of code. The header file and functions along with syntax and purpose are described completely to make it understandable to the reader. The KPs role regarding the project is explained in detail in section 4.7. The project weather station has been run and simulated using Opnet software 17.1 that has the facility of supporting the interface control information. The results obtained are according to the expectations and they proof that the scope of the project as a real time application.

Chapter five presented the results of WeS simulation. There can be three cases which are luminous threats, temperature threats, and humidity threats. As all these three cases are set in the contents of interface control information, therefore after running the simulation results have been shown in figures in section 5.2 and subsequent subsections These cases will

remain the same throughout other results because their values are already set. The message is printed in the console without error which shows the simulation is compiled successfully. In different cases WeS behaviour is accurate showing the right messages in the console that verifies the design. At the end, section 5.3, some limitations of the project are described showing that much more work is left in this field but of which will contribute the new knowledge of this thesis. For example project is limited for certain operations because it does not analyse the network for some statistics. Due to model limitations for WiMAX, the following features have not been implemented that includes network-assisted initial ranging during handover; Base station-initiated periodic ranging. It is not possible to perform the frequency division duplex mode. However project opens the new lines of thoughts of speculations to work in these areas.

Section 6.2 provides new concepts for future research work in DSP Extra-tropical Cyclones Warning System.

6.2 Future Work

The Opnet is leading research tool being used for advanced networking systems.

It can be used to build any kind of simulation scenarios for example for the deployment of communication systems that are complex. They can be designed keeping in view network managing or quality of service assurance issues. As in this project WiMAX technology is used, using the same technology code can be designed that will govern the bandwidth allocation so that the different users will have same opportunity of transmitting data in WiMAX network for voice, video conferencing and other applications.

With slight modifications, WeS model can be used for defence purpose. The system should work as continuous monitoring if there is any threat of enemy attack and early warning message will be generated that will be transmitted to WiMAX base station and from base station to department of defence. In this way one can be secured of not only from natural disasters but also from enemy attacks.

Radars can be replaced by the WiMAX technology by deploying WiMAX base stations and WiMAX based satellites which can be used to control air traffic (defence and civil) and submarines control and hence further defence.

Advance public transport scheduling system based on WiMAX network can be designed which provides a precise arrival time to the station as bus can be delayed by the traffic jams and due to the bad weather condition.

References

Abusch-Magder et al., 2007 Abusch-Magder, D., Bosch, P., Klein, E, T., Polakos, A, P., Samuel, G, L and Viswanathan, H., (2007). 911-NOW: A Network on Wheels for Emergency Response and Disaster Recovery Operations. Bell Labs Technical Journal 11(4). pp 113 – 133.

AD7714, 1998 Analog Devices, Inc., AD7714: CMOS, 3V/5V, 500 µA, 24-Bit Sigma-Delta, Signal Conditioning ADC Data Sheet (Rev C, 06/1998), (1998). USA.

Ames, 2008 Ames, R. (2008). SUIRG, Inc. Adam Edwards, SES-NewSkies/SUIRG Kenneth Field Test Report WiMAX Frequency Sharing with FSS Earth Stations Carrigan. USA: Users Interference Reduction Group, In.

Andrews et al., 2007 Andrews, J. G., Ghosh, A & Muhamed, R., (2007). Fundamentals of WiMAX, Understanding Broadband Wireless Networking. USA: Prentice-hall, Pearson Education, Inc.

Application Note, 2006 Application Note, (2006). A Solar-Powered WiMAX Base Station Solution" Intel Netstructure® WiMax Baseband Card. USA: Intel Corporation

CAP, 2010 OASIS, (2010). Common Alerting Protocol Version 1.2, USA: Available from < http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.htm > [Accessed 10 July, 2010].

Deruyck, 2010 Deruyck, M, (2010), Comparison of power consumption of mobile WiMAX, HSPA and LTE access networks, Dept. of Inf. Technol., Ghent Univ., Ghent, Belgium, Telecommunications Internet and Media Techno Economics (CTTE)

ERRICSON, 2003 ERRICSON, (2003). WLAN In Disaster and Emergency Response (WIDER). Available from: < http://www.itu.int/itudoc/itu-t/workshop/ets/s5p4.pdf> [Accessed March, 2010].

Fazel, 2008 Fazel, K and Kaiser, S, (2008). Multi-Carrier and Spread Spectrum Systems: From OFDM and MC-CDMA to LTE and WiMAX, 2nd Edition, John Wiley & Sons, ISBN 978-0-470-99821-2

Flickenger et al., 2007 Flickenger, R., Aichele, c., Büttrich, S., Drewett, M, L., Escudero-Pascual, A., Berthilson, L., Fonda, C., Forster, J., Howard, I., Johnston, K., Krag, T., Kupfermann, G., Messer, A., Neumann, J., Pietrosemolim, E., Renet, F and Zennaro, M., (2007). Wireless Networking in the Developing World. 2nd ed. Hacker Friendly LLC

IEEE, 2004 IEEE Computer Society and IEEE Microwave Theory and Techniques Society, (2004). IEEE Standard for Local and Metropolitan Area Networks Part 16: Air interface for Fixed Broadband Wireless Access Systems. USA, IEEE.

IEEE, 2005 IEEE Computer Society and IEEE Microwave Theory and Techniques Society, (2005). IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems. USA, IEEE.

Nuaymi, 2007 Nuaymi, L., (2007). WiMAX : Technology for Broadband Wireless Access. USA: John Wily & Sons Inc.

Life of Guangzhou, 2009 Life of Guangzhou, (2009). Guangzhou Wireless City Project to Adopt TD-SCDMA+WLAN Standard, Available from: < http://www.lifeofguangzhou.com/node_10/node_34/node_280/2009/07/22/124824162567597.shtml > [Accessed 22nd July 2009] Beijing: Interactive Information Network Co., Ltd.

Nauymi, 2007 Loutfi Nuaymi, 2007. WiMAX Technology for Broadband Wireless Access, West Sussex: Wiley.

Opnet 17.1, 2012 Opnet, Inc., (2012). C:/OPNET/17.1.A/doc/modeler/wwhelp/wwhimpl/js/html/wwhelp.htm#href=Models/WiMAX_Model_desc.42.03.html#750715

SDMA, 2010 SDMA.,(2010). Situation Report No. 5 (30th July 2010) of Monsoon Rains / Floods in AJ&K. AZAD GOVERNMENT OF THE STATE OF

JAMMU & KASHMIR STATE DISASTER MANAGEMENT AUTHORITYMUZAFFARABAD.

Stallings, 2004 Stallings, W., (2004). Wireless Communications and Networks. 2nd ed. USA: Prentice-Hall, Inc.

WiMAX-Communication, 2011 http://www.wimax-com.com/site/index.php?page_id=460

Zhang, 2008 Zhang, Y and Chen H, H., (2008). Eds, MOBILE WiMAX. Auerbach.

Appendices

Appendix A: NetDoctor Report

OPNET NetDoctor

Table of Contents

Executive Summary

Score: **100** (good)
Previous Score: **100**
Difference: **0**

This NetDoctor report shows the state of the network named "**CloneWAN_V6**". The data used to generate this report came from 1 tested device and 154 rules. The score for this report is 100 (out of 100).
No reported issues were found in this network.

Additionally, this report was compared to the NetDoctor report "**NetDoctor Report**" generated Thursday, May 5, 2011. However, there were no new reported issues.

Summaries

Advisories: CA 2003 23 RPCSS Vulnerabilities in Windows
Advisories: CA 2003 28 Buffer Overflow in Windows Workstation Service
Advisories: CA 2004 01 Multiple H323 Message Vulnerabilities

This device-centric detailed report was generated using:
Project: CloneWAN_V6
Scenario: scenario1

Aborted Rules

AAA: Verify Accounting Configuration

This rule verifies that AAA accounting is configured as specified on each device.

NOTE: This rule operates on Cisco devices running IOS only.

Sources:
- 4.6.2, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 13, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 10.2, 10.3, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- AC-13, AU-2, AU-3, AU-10, IR-5, Special Publication 800-53, National Institute of Standards and Technology.
- 8.1.1, 10.1.2, 10.1.3, 10.10.1, 10.10.2, 10.10.4, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(3)(ii)(A), 164.308(a)(3)(ii)(A), 164.308(a)(5)(ii)(C), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

To check the accounting configuration of devices, specify an input file for the parameter Accounting Configuration. Add lines in the following format to the file:

<Accounting Activity or Command Level>,<Method List>,<Event>,<tacacs+|radius|AAA Server Group(s)>

- Valid Accounting Activities are: auth-proxy|connection|exec|network|resource|system
- Valid Events are: none|start-stop|stop-only|wait-stop|start-stop-failure|stop-failure

E.g., exec,default,start-stop,tacacs_servers

- If less than four values are specified on a line, the remaining will be considered unapplicable for the account type.
- If more than four values are specified on a line, the additional values will be ignored.

ATM: Verify Administrative Cost

This rule checks whether the specified Administrative Cost is set on each ATM NNI port in the network.

NOTE: The Administrative Cost should be specified as a positive integer.

Administration: Verify Blocked Incoming Services

This rule checks if the virtual lines on each device running Cisco IOS in the network, are blocking the specified incoming services.

NOTE: Virtual lines that block more services than those specified are not reported.

Sources:
- 3.2.2, 3.4.3, 3.4.4, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 6, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- NET0680, NET0740, Network Infrastructure STIG (Version 6, Release 4), Defense Information Systems Agency.
- 2.2, 2.2.2, 2.3, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- AC-17, CM-7, Special Publication 800-53, National Institute of Standards and Technology.
- 10.1.2, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Administration: Verify Blocked Outgoing Services

This rule checks if the virtual lines on each device running Cisco IOS in the network, are blocking the specified outgoing services.

NOTE: Virtual lines that block more services than those specified are not reported.

Sources:
- 3.2.2, 3.4.3, 3.4.4, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 6, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 2.2.2, 2.3, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- CM-7, Special Publication 800-53, National Institute of Standards and Technology.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Administration: Verify Device OS Versions

This rule checks if the devices in the network are running the specified version of their operating systems.

NOTE: This rule checks the OS version on Cisco IOS, Cisco Switch IOS, Cisco CatOS, Cisco PIX, Juniper, Extreme, Foundry, Check Point, and Alcatel-Lucent devices.

Sources:
- 4.5.1, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 3, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 6.1, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- 10.1.2, 12.4.1, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Parameter Usage:
You can specify the Vendor model, OS Type and the permitted/forbidden OS versions as values of parameters or in an XML specification file. When directly specifying values of the parameters you can check for a maximum of 4 OSes.

Using the XML specification file allows you to check for an unlimited number of OSes.

XML specification file format:

```
<VerifyDeviceOSVersions version="2">
  <DeviceOSSpecification>
    <!-- One for each Vendor model to be tested -->

    <VendorModels>model</VendorModels>
      <!-- The semi-colon separated list of vendor models (regular expressions) whose OS type and OS versions you
wish verify.
           (This tag is optional and more than one VendorModels tags can be
           specified for each DeviceOSSpecification section.

           If specified devices of these vendor model(s)
           are verified that they are running of the specified OS types) -->

    <DeviceNameRegex>(?i)nyc.*</DeviceNameRegex>
      <!-- Device name regex, this Device OS specification is only tested on devices whose name matches this regex.
           (Optional tag, the default value is ".*", multiple DeviceNameRegex tags can be included.) -->

    <OSVersions>
      <!-- Includes the OS version specifications (This tag is mandatory, one per DeviceOSSpecification section). -->

      <OSVersion>
        <!-- The OS version specification (This tag is mandatory). If VendorModels tag has been specified then one
or more OS version specification sections can be included. However, if VendorModels tag has not been specified then
only one OS version section can be included. -->


          <OSType>os_name</OSType>
            <!-- The OS whose versions you wish to verify (This tag is mandatory).
                 One and only one OSType tag should be included for each OSVersion section.

                 The OSType should be one of the following:

                 Cisco IOS
                 Cisco Switch IOS
                 Cisco CatOS
                 Cisco PIX
                 Cisco ASA
                 Cisco FWSM
                 Cisco IOS-XR
                 Juniper JUNOS
                 Juniper ScreenOS
                 ExtremeWare
                 Foundry IronWare
                 Checkpoint
                 Alcatel AOS
                 Alcatel AOS-W
                 Alcatel OmniStack OS
                 Alcatel-Lucent SR OS
    -->

          <PermittedVersions>os_version</PermittedVersions>
            <!-- One permitted OS version or a semi-colon separated list of permitted OS versions.
                 Multiple PermittedVersions tags can be included.

                 Devices (identified by OSType or VendorModel) must be running one of the permitted OS versions.
                 (This tag is optional, however, either this or ForbiddenVersions tag must be specified
                 for each OSVersion) -->
```

```
        <ForbiddenVersions>os_version</ForbiddenVersions>
          <!-- One forbidden OS version or a semi-colon separated list of forbidden OS versions.
               Multiple ForbiddenVersions tags can be included.

               Devices (identified by OSType or VendorModel) must not be running any of the forbidden OS
versions.
               (This tag is optional, however, either this or PermittedVersions tag must be specified
               for each OSVersion) -->

        <UseRegExMatching>Yes or No</UseRegExMatching>
          <!-- "Yes" if statements in the permitted or forbidden versions are to be treated as regular expressions. "No"
otherwise.
               (Optional tag, the default value is "Yes")-->

        <Severity>ERROR</Severity>
          <!-- The severity of report entries generated by this rule if violations are found.
               (Optional tag, the default value is "WARNING"). The Severity should be one of the following:

               ERROR
               WARNING
               NOTE
          -->

        </OSVersions>
      </OSVersion>
  </DeviceOSSpecification>
</VerifyDeviceOSVersions>
```

Administration: Verify Exec Banner

This rule verifies that the exec banner on each device is configured as specified.

NOTE:
- This rule operates on Cisco IOS, PIX, FWSM, and ASA devices.
- In addition to simple strings, the message can be specified as Python regular expressions.
- Newline characters in the specified and configured message are ignored
- Do not include "^C" in the specified IOS message

Sources:
- 4.1.5, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 5, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- AC-8, Special Publication 800-53, National Institute of Standards and Technology.
- 11.5.1, ISO-17799, Code of Practice for Information Security Management.

Parameter Usage:
- To skip checking a particular operating system, set that operating system's "Match Criteria" parameter to "Do Not Check"
- To check that a particular operating system has a specific banner message, set that operating system's "Match Criteria" parameter to "Exact Match" and specify the file containing the required message in the "Message" parameter.
- To check that a particular operating system has a banner message that begins with specific text, set that operating system's "Match Criteria" parameter to "Starts With" and specify the file containing the required message in the

"Message" parameter.
- To check that a particular operating system has any banner message configured, set that operating system's "Match Criteria" parameter to "Any"

Administration: Verify Hostnames

This rule verifies that the devices in the network have a permitted hostname and not one of the forbidden hostnames.

Parameter Usage:
- At least one of the Permitted or Forbidden Hostnames parameters must be set
- If you specify only permitted hostnames, then all device hostnames must match one of the permitted hostnames
- If you specify only forbidden hostnames, then no device hostnames can match any of the forbidden hostnames

Administration: Verify Login Banner

This rule verifies that the login banner on each device is configured as specified.

NOTE:
- This rule operates on Cisco IOS, PIX, FWSM, and ASA devices.
- In addition to simple strings, the message can be specified as Python regular expressions.
- Newline characters in the specified and configured message are ignored
- Do not include "^C" in the specified IOS message

Sources:
- 4.1.5, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 5, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- AC-8, Special Publication 800-53, National Institute of Standards and Technology.
- 11.5.1, ISO-17799, Code of Practice for Information Security Management

Parameter Usage:
- To skip checking a particular operating system, set that operating system's "Match Criteria" parameter to "Do Not Check"
- To check that a particular operating system has a specific banner message, set that operating system's "Match Criteria" parameter to "Exact Match" and specify the file containing the required message in the "Message" parameter.
- To check that a particular operating system has a banner message that begins with specific text, set that operating system's "Match Criteria" parameter to "Starts With" and specify the file containing the required message in the "Message" parameter.
- To check that a particular operating system has any banner message configured, set that operating system's "Match Criteria" parameter to "Any"

Administration: Verify MOTD Banner

This rule verifies that the message of the day (MOTD) banner on each device is configured as specified.

NOTE:
- This rule operates on Cisco IOS, PIX, FWSM, and ASA devices.
- In addition to simple strings, the message can be specified as Python regular expressions.
- Newline characters in the specified and configured message are ignored
- Do not include "^C" in the specified IOS message

Sources:
- 4.1.5, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 5, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- AC-8, Special Publication 800-53, National Institute of Standards and Technology.
- 11.5.1, ISO-17799, Code of Practice for Information Security Management.

Parameter Usage:
- To skip checking a particular operating system, set that operating system's "Match Criteria" parameter to "Do Not Check"
- To check that a particular operating system has a specific banner message, set that operating system's "Match Criteria" parameter to "Exact Match" and specify the file containing the required message in the "Message" parameter.
- To check that a particular operating system has a banner message that begins with specific text, set that operating system's "Match Criteria" parameter to "Starts With" and specify the file containing the required message in the "Message" parameter.
- To check that a particular operating system has any banner message configured, set that operating system's "Match Criteria" parameter to "Any"

Administration: Verify OS Images

This rule verifies that the devices in the network are running one of the permitted operating system images and not one of the forbidden operating system images.

NOTE: This rule operates on Cisco devices running IOS only.
Sources:
- 3.3.2, 4.1.2, 4.5.5, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 3, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 6.1, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- 12.4.1, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Administration: Verify Permitted Incoming Services

This rule checks if the virtual lines on each device running Cisco IOS in the network, are permitting the specified incoming services.

NOTE: This rule reports virtual lines that do not permit the same services as those specified in the rule parameter.

Sources:
- 3.2.2, 3.4.3, 3.4.4, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 6, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- NET0680, NET0740, Network Infrastructure STIG (Version 6, Release 4), Defense Information Systems Agency.
- 2.2, 2.2.2, 2.3, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- AC-17, CM-7, Special Publication 800-53, National Institute of Standards and Technology.
- 10.1.2, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Administration: Verify Permitted Outgoing Services

This rule checks if the virtual lines on each device running Cisco IOS in the network, are permitting the specified outgoing services.

NOTE: This rule reports virtual lines that do not permit the same services as those specified in the rule parameter.

Sources:
- 3.2.2, 3.4.3, 3.4.4, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 6, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 2.2.2, 2.3, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- CM-7, Special Publication 800-53, National Institute of Standards and Technology.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Administration: Verify Summer Time Clock Settings

Correlating log files of devices may be difficult if the time stamps on the log messages have to be adjusted for different summer time clock settings. This rule verifies that the device is configured with the specified summer time clock settings.

NOTE:
- This rule operates on Cisco IOS, PIX, and ASA devices.
- On PIX 7.x and ASA devices in multiple context mode, this rule only operates on the System context.

Sources:
- 4.5.2, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 12, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 10.4, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- AU-8, Special Publication 800-53, National Institute of Standards and Technology.
- 10.10.6, ISO-17799, Code of Practice for Information Security Management.

This rule could not be run because the parameters to check device operating systems were set to "No".

Administration: Verify TCP Synwait Time

This rule verifies that the device is configured with the specified TCP Synwait Time value.

Setting the TCP Synwait Time is useful in defeating SYN flooding attacks, a form of denial-of-service (DoS) attack.

NOTE: This rule operates on Cisco devices running IOS only.

Sources:
- 8, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 6.5.9, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- SC-5, Special Publication 800-53, National Institute of Standards and Technology.

This rule was aborted because the parameter "Synwait Time" was set to a value less than 5 seconds.

Administration: Verify Time Zone Settings

Correlating log files of devices may be difficult if the time stamps on the log messages have to be adjusted for different time zone settings. This rule verifies that the device is configured with the specified time zone settings.

NOTE:
- This rule operates on Cisco IOS, PIX, and ASA devices.
- On PIX 7.x and ASA devices in multiple context mode, this rule only operates on the System context.

Sources:
- 4.5.2, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 12, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 10.4, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- AU-8, Special Publication 800-53, National Institute of Standards and Technology.
- 10.10.6, ISO-17799, Code of Practice for Information Security Management.

This rule could not be run because the parameters to check device operating systems were set to "No".

Administration: Verify Well-Known Passwords Not Used for Telnet Access

Well-known passwords should not be used for telnet access to a device. Attackers may easily guess well-known passwords and gain access to a device.

NOTE:
- This rule operates on Cisco PIX, ASA and FWSM devices.
- On PIX 7.x and ASA devices in multiple context mode, this rule only operates on non-System contexts.
- The parameter file containing encrypted passwords should have one encrypted password per line. Leading and trailing spaces on each line are ignored.

Sources:
- 4.1.5, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 4, 5, Cisco IOS Switch Security Configuration Guide (Version 1.0), National Security Agency.
- 2.2, 8.5.8, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- 11.3.1, 11.5.3, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(4)(ii)(B), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

This rule was aborted as the "Encrypted Passwords File" parameter is set to "<Not Set>".

Voice over IP: Verify H225 Concurrent Calls

Verify that an H.323 gateway has the specified number as the allowed H.225 concurrent calls per TCP connection.

NOTE: If the rule parameter "Concurrent Calls" is set to its default value of -1, then this rule will not perform its check.

Voice over IP: Verify Zone Prefix Count

Verify that an H.323 Gatekeeper has no more than the specified number of zone prefixes.

NOTE: If the rule parameter "Max. Zone Prefixes" is set to its default value of -1, then this rule will not perform its check.

Wireless LAN: Encryption Not Enabled

This rule verifies that encryption has been enabled for communication between the access point and wireless clients associated with it. If not enabled, this communication can be accessed by other wireless devices for malicious purposes.

NOTE: This rule operates on Cisco Aironet Access Points running IOS only.

Sources:
- 4.1.1, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- AC18, SC8, SC9, SC14, Special Publication 800-53, National Institute of Standards and Technology.
- 10.6.1, 10.6.2 ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(4)(ii)(A), 164.312(e)(2)(ii), 164.312(a)(2)(iv), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Wireless LAN: Verify Access Point Encryption Mode

This rule verifies that the encryption mode of access points is as specified.

NOTE: This rule operates on Alcatel-Lucent OmniAccess devices and Cisco Aironet Access Points running IOS.

Sources:
- AC-18, Special Publication 800-53, National Institute of Standards and Technology.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(4)(ii)(A), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

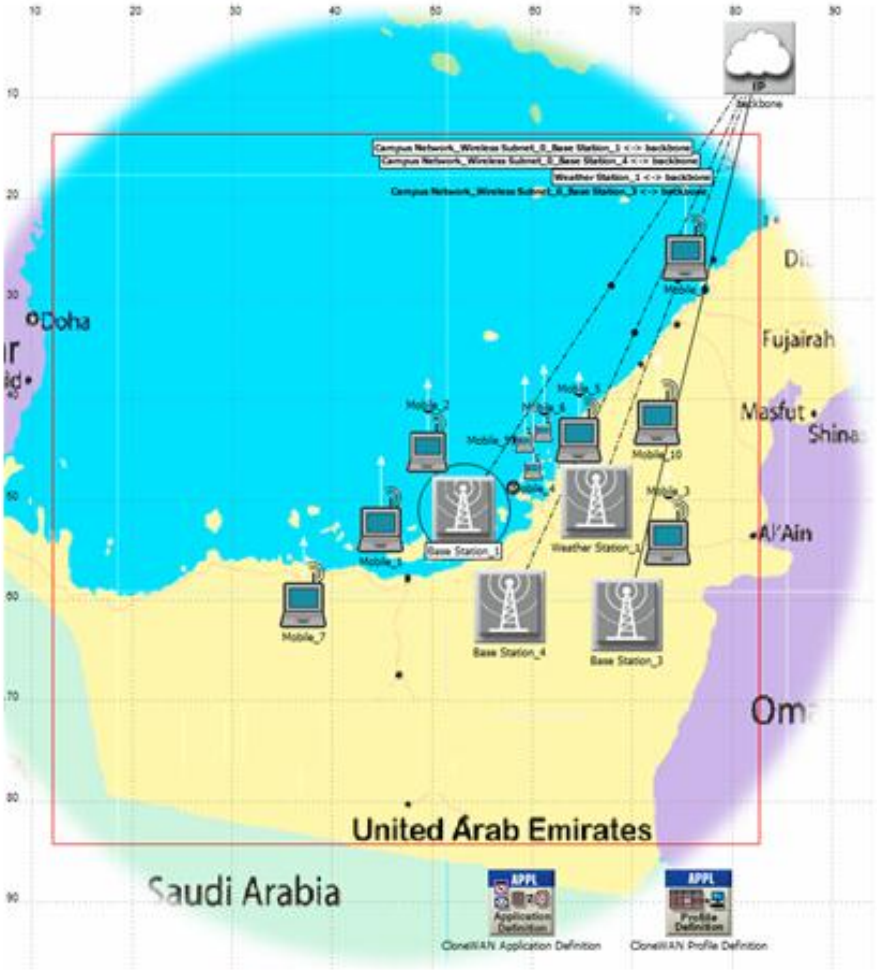
Wireless LAN: WEP Encryption is Optional

This rule verifies that the access point does not allow clients to use WEP encryption optionally. All communication should be encrypted and an access point should enforce encryption rather than letting the client choose whether or not to use encryption.

NOTE: This rule operates on Cisco Aironet Access Points running IOS only.

Sources:
- 2.2, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- AC18, SC8, SC9, SC14, Special Publication 800-53, National Institute of Standards and Technology.
- 10.6.1, 10.6.2 ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(4)(ii)(A), 164.312(e)(2)(ii), 164.312(a)(2)(iv), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Network Diagram



Differences Summary

This report was compared to:

- Template: Default NetDoctor Report
- Title: NetDoctor Report
- Project: CloneWAN_V3
- Scenario: scenario1
- Generated: Thursday, May 5, 2011 at 23:56
- Threshold: Errors, Warnings, and Notes

Rules run: 154 (from 7 rule suites)
New Errors: 0
New Warnings: 0
New Notes: 0
No Longer Detected: 0

Summaries

Advisories: CA 2003 23 RPCSS Vulnerabilities in Windows

Vulnerabilities in Microsoft Windows may allow an attacker to execute arbitrary code with system privileges or cause a denial of service. Certain port-protocol pairs can be blocked on routers and firewalls to reduce the chances of successful exploitation. This rule checks if the specified port-protocol pairs are blocked on the active interfaces of devices in the network.

This rule is based on the rule "Security: Interfaces With Unblocked Ports". The default port-protocol pairs specified are those listed in CERT Advisory: CA-2003-23.

Sources:
- CERT Advisory CA-2003-23 RPCSS Vulnerabilities in Microsoft Windows
- 3.2.2, 3.2.3, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 2.2, 6.5.9, 6.6, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- RA-5, Special Publication 800-53, National Institute of Standards and Technology.
- 12.4.1 ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(5)(ii)(A), 164.308(a)(5)(ii)(A), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Checked port-protocol pairs:
135/tcp, 135/udp, 137/udp, more

Network Summary	
Devices	1
Interfaces	0
Compliant Interfaces	0
Partially Compliant Interfaces	0
Non-Compliant Interfaces	0
Interfaces With Undefined Filters	0
Interfaces Without Filters	0
Interfaces With Non-Compliant Filters	0

Device Summary							
Device	Interfaces	Compliant Interfaces	Partially Compliant Interfaces	Non-Compliant Interfaces	Interfaces With Undefined Filters	Interfaces Without Filters	Interfaces With Non-Compliant Filters

Advisories: CA 2003 28 Buffer Overflow in Windows Workstation Service

A buffer overflow vulnerability in Microsoft's Windows Workstation Service may allow an attacker to execute arbitrary code or cause a denial of service. Certain port-protocol pairs can be blocked on routers and firewalls to reduce the chances of successful exploitation. This rule checks if the specified port-protocol pairs are blocked on the active interfaces of devices in the network.

This rule is based on the rule "Security: Interfaces With Unblocked Ports". The default port-protocol pairs specified are those listed in CERT Advisory: CA-2003-28.

Sources:
- CERT Advisory CA-2003-28 Buffer Overflow in Windows Workstation Service
- 3.2.2, 3.2.3, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 2.2, 6.5.9, 6.6, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- RA-5, Special Publication 800-53, National Institute of Standards and Technology.
- 12.4.1, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(5)(ii)(A), 164.308(a)(5)(ii)(A), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Checked port-protocol pairs:
138/tcp, 138/udp, 139/tcp, more

Network Summary	
Devices	1
Interfaces	0
Compliant Interfaces	0
Partially Compliant Interfaces	0
Non-Compliant Interfaces	0
Interfaces With Undefined Filters	0
Interfaces Without Filters	0
Interfaces With Non-Compliant Filters	0

Device Summary							
Device	Interfaces	Compliant Interfaces	Partially Compliant Interfaces	Non-Compliant Interfaces	Interfaces With Undefined Filters	Interfaces Without Filters	Interfaces With Non-Compliant Filters

Advisories: CA 2004 01 Multiple H323 Message Vulnerabilities

Vulnerabilities in the H.323 protocol may allow an attacker to execute arbitrary code or cause a denial of service. Certain port-protocol pairs can be blocked on routers and firewalls to reduce the chances of successful exploitation. This rule checks if the specified port-protocol pairs are blocked on the active interfaces of devices in the network.

This rule is based on the rule "Security: Interfaces With Unblocked Ports". The default port-protocol pairs specified are those listed in CERT Advisory: CA-2004-01.

Sources:
- CERT Advisory CA-2004-01 Multiple H.323 Message Vulnerabilities
- 3.2.2, 3.2.3, Router Security Configuration Guide (Version 1.1), National Security Agency.
- 2.2, 6.5.9, 6.6, PCI Data Security Standard (Version 1.1), PCI Security Standards Council.
- SAFE: A Security Blueprint for Enterprise Networks; Cisco Systems, Inc.
- RA-5, Special Publication 800-53, National Institute of Standards and Technology.
- 12.4.1, ISO-17799, Code of Practice for Information Security Management.
- 164.306(a)(2), 164.308(a)(1)(ii)(B), 164.308(a)(5)(ii)(A), 164.308(a)(5)(ii)(A), Health Insurance Reform: Security Standards; Final Rule (Feb. 20, 2003), Department of Health and Human Services.

Checked port-protocol pairs:
1720/tcp, 1720/udp

Network Summary	
Devices	1
Interfaces	0
Compliant Interfaces	0
Partially Compliant Interfaces	0
Non-Compliant Interfaces	0
Interfaces With Undefined Filters	0
Interfaces Without Filters	0
Interfaces With Non-Compliant Filters	0

Device Summary							
Device	Interfaces	Compliant Interfaces	Partially Compliant Interfaces	Non-Compliant Interfaces	Interfaces With Undefined Filters	Interfaces Without Filters	Interfaces With Non-Compliant Filters

Additional Data

Checked port-protocol pairs:

135/tcp
135/udp
137/udp
138/udp
139/tcp
445/tcp
445/udp
539/tcp

Back

Checked port-protocol pairs:

138/tcp
138/udp
139/tcp
139/udp
445/tcp
445/udp

Back

Appendix A: All Devices

Campus Network.Wireless Subnet_0.Base Station_1 *(passed)*
Campus Network.Wireless Subnet_0.Base Station_3 *(passed)*
Campus Network.Wireless Subnet_0.Base Station_4 *(passed)*
Campus Network.Wireless Subnet_0.CloneWAN Application Definition *(passed)*
Campus Network.Wireless Subnet_0.CloneWAN Profile Definition *(passed)*
Campus Network.Wireless Subnet_0.Weather Station_1 *(passed)*
Campus Network.Wireless Subnet_0.backbone *(passed)*
Mobility Config *(passed)*
WiMAX_Config *(passed)*

Appendix B: Report Score Breakdown

Score by Device

100 Campus Network.Wireless Subnet 0.Base Station 1

Appendix C: Breakdown of Device Scores

Score by Rule

Campus Network.Wireless Subnet_0.Base Station_1

Rule	Reported Issues			Raw Score	Weighting				Contribution to Device Score
	Errors	Warnings	Notes	Rule Weight	Severity Weight	Total Weight		Relative Weight	
Administration: Line Password Not Set	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify Authentication Failure Rate	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify Minimum Password Length	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify Outgoing Sessions Disabled	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify SSH Timeout and Authentication Retries	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify SSH and Telnet Access Control	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify Timeout for Login Sessions	0	0	0	100	10	10	100	0.1111	11.11
Administration: Verify VTY Access Control List	0	0	0	100	10	10	100	0.1111	11.11
HTTP: Verify HTTP Access Control	0	0	0	100	10	10	100	0.1111	11.11
Device Score									100

Appendix D: Breakdown of Rule Scores

Score by Rule

Administration: Line Password Not Set

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify Authentication Failure Rate

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify Minimum Password Length

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify Outgoing Sessions Disabled

Average Score Across All Devices

Device	Reported Issues	Raw	Weighting	Contribution to

	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify SSH Timeout and Authentication Retries

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify SSH and Telnet Access Control

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify Timeout for Login Sessions

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

Administration: Verify VTY Access Control List

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100

HTTP: Verify HTTP Access Control

Average Score Across All Devices

Device	Reported Issues			Raw Score	Weighting		Contribution to Rule Score
	Errors	Warnings	Notes		Severity Weight	Relative Weight	
Campus Network.Wireless Subnet_0.Base Station_1	0	0	0	100	10	1	100
Rule Score							100