DEVELOPMENT OF EXPERIMENTAL FRAME AND ABSTRACT DEVS MODELS TO SUPPORT SCOPE NETWORK EXPANSION

by

Mahesh Veena

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN COMPUTER ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

2005

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: ________________________________

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

____________________________________ ________________________

Bernard P. Zeigler Date
Professor of Electrical and Computer Engineering

ACKNOWLEDGEMENTS

I would like to express my thanks and gratefulness to Professor Bernard Zeigler, my thesis advisor. His invaluable advice and suggestions helped me greatly in completing this study. He gave me the freedom to work on topics that interested me. I also thank him for financially supporting me during my study at the University of Arizona. His constant support and encouragement continue to inspire and motivate me.

I thank Professor Jerzy Rozenblit for serving on my thesis defense committee and his suggestions to help improve the work. A special thanks to Dr. Moon Ho Hwang who made changes to his schedule to be on my committee. His advice and help in generating results and improving the content of the work are greatly appreciated.

I have a wonderful family to thank for their help. I thank my parents, for their many sacrifices and their continuous encouragement, motivation, support and belief in me. My father's dedication to constant learning, research and improvement has motivated me greatly and will continue to do so. My mother's integrity and modesty have inspired me always. A special thanks to my sister and brother-in-law who also encouraged me greatly to start and complete my study. Thanks to them for the great vacations I spent with them and their support through out this endeavor. Thanks also to my uncle and aunt in Boston for their constant encouragement.

Thanks to the members of the ACIMS lab. Thanks to Dr. Phil Hammonds who helped us in working on this project at Northrop Grumman. A special thanks to Chungman and Saehoon, many a time have they stayed back in order to take me home after a long day's work. Thanks to Saurabh who worked with me on this project. Thanks to Brett, Eddie, Fahd, Greg, Raj, Swami and John Lee for creating a fun environment in the lab. Thanks also to Dr. Doohwan Kim, Jim and Xiaolin who gave me helpful advice. Thanks to Stephen for helping me generate the figures and animations for my presentation.

Thanks to Mike Skutt, Anthony Stamford, Joe Schulte and Danny Hurd at Northrop Grumann for helping us work on this project and giving us information whenever we needed it. Thanks to Dr. Eric Johnson at NMSU who took time off his schedule to help us understand NetSim. Thanks to Mr. Jay Roberts and Mr. Woodring from the SCOPE Command project for their helpful suggestions.

Along the way, I met and made many awesome friends. There are too many to name here. In particular, sincere thanks to Walter and Saskia for always being there and their wonderful friendship. Thanks also to Avinash, Chetan, Jens, Neema, Nikhil and Rajesh.

A special thanks to Dr. H.S.Nagaraj, Mr. Prakash and Mr. Aithal who took great pains to help me start off on this wonderful journey.

Thanks to coffee, food and music – faithful companions for ever!

To my parents and sister: M, N and M

TABLE OF CONTENTS

TABLE OF CONTENTS - continued

TABLE OF CONTENTS - continued

LIST OF CHARTS AND FIGURES

LIST OF CHARTS AND FIGURES - continued

LIST OF TABLES

Abstract

Computer and communication network administrators or designers face a constant need to increase the size of the network or reconfigure the existing network. Whenever there is any reconfiguration involved, they need to first test whether the new configuration works. Online testing is almost impossible in present day networks due to sheer volume and usage. Offline testing using computer models and simulations seems to be a feasible solution for such administrators or designers. In this thesis, the Discrete Event System Specification (DEVS) is proposed as a suitable tool to build such models and run simulations to yield approximate results.

In this research, the US Air Force SCOPE (Systems Capable of Planned Expansion) Command system is taken up as the test network. Basically, the system consists of ground stations which are equipped with radios capable of implementing the Automatic Link Establishment (ALE) Protocol. There are aircraft moving around in the ionosphere that also carry such ALE radios. Communication takes place between an aircraft and a ground station. Once an ALE radio is in a link with another radio, it cannot be used for any other connections. Hence with a large number of aircraft, the probability of a ground station being unavailable is quite high. In addition to this, if the SNR of a signal is not good enough, then a link cannot be established. The main interest of the designers of the SCOPE Command system lies in finding out how many calls can be linked or dropped based on these and some other factors. Rather than being concerned with the modeling of the ALE protocol to its lowest level of details, models at an appropriate level of abstraction are employed to test various scenarios varying the number and location of

ground stations, the number of ALE radios at each ground station, the number of aircraft, the SNR thresholds and so on. Such models are compared against an existing high resolution model and indicative results are reported. An Experimental Frame is developed to facilitate the design and conduction of experiments which can help in comparing the performance of various system designs.

1. Introduction

As computer networks and communication systems grow in size, it becomes more challenging to study the network performance characteristics. Expansion of networks will involve careful planning. Crucial questions such as which set of resources are crucial to network operation, the amount of utilization of existing resources and so on need to be answered. These answers would help in deploying additional resources. Then come questions such as what kinds of resources are needed, where are they to be deployed and so on.

Once the proportion of resources being used in a network is known, either through data collection or through heuristics, any additional demands for network service may be suitably met. For example, when the shortest path for routing of messages from source to destination is already in use and cannot accommodate any more traffic, alternate routes may be used. These alternate routes can be selected on various criteria.

Network utilization and performance studies may be carried out in real time. That is, the required data can be collected while the network is actually in operation. But the effects of additional resources on the network, alternate routing schemes and other modifications to the system cannot usually be studied while the network is in use. In such studies, modeling and simulation play a very important part role. Models of the various components are built and then simulations are run as needed. The results gathered from these simulations may then be used suitably.

In this research effort the performance of the Systems Capable of Planned Expansion (SCOPE) Communication System is studied. Basically it is a communication system utilizing the HF radio waves and is employed by the US Army. It has its own communication infrastructure and protocols, which are explained in detail in chapter two. It is not practicable to experiment with the actual system to identify the effects of adding or removing radios and other resources. Further, the network simulators that are available at present commercially are not well equipped to answer several important questions of the SCOPE Command system planners/users.

In order to study SCOPE system, the Discrete Event System Specification, (DEVS) modeling and simulation formalism is used. DEVS provides the modeler with many advantages. Chapter three explains the DEVS formalism in greater detail. The focus in this study is more on developing abstract models with just enough features to answer the questions the designers of the system have rather than developing models with very high resolution in details.

Once the models are built, their performance has to be analyzed. For this purpose, various experiments need to be generated. In modeling and simulation, this is the responsibility of the Experimental Frame (EF). The Experimental Frame is capable of generating different experiments needed to evaluate the system performance. In the EF, various scenarios can be set up and the performance measures to be collected can also be specified.

1.1. Motivation

The motivation behind this research is to explore the use of the DEVS Modeling and Simulation formalism as a tool in studying system performance, especially as applied to computer and/or communication networks. Though there are some standard simulators in the market such as OPNET, NS2, etc, all these simulators have some drawbacks as will be pointed out later. DEVS can be used as a replacement for these tools. Its inherent discrete nature and also its aid in creating models with the required amount of detail provide powerful advantages over the other simulators.

There are three general approaches to system simulation: emulation, quantum simulation and directed simulation [1]. In *emulation*, the system is modeled to its last set of specification detail. With this approach, however, model development and simulation studies consume more time. This level of detail may be useful in studies where the primary concern is to check whether the system works or not. *Quantum Simulation*, on the other hand, strips away all the details and only considers abstract descriptions of the system behavior. This relies on random variables and models the bulk behavior of the system. With such an approach, alternate scenarios can be studied quickly and easily. At the very least, approximate values for system performance can be found. Gradually more details could be added depending on the level of accuracy needed. *Directed Simulation* uses a middle of the road approach. In such a simulation, one or a few aspects of the system are modeled in detail and others are abstracted. Adding details to the quantum simulation would usually lead to a directed simulation.

In the domain of networks and communication systems, standard simulators are available such as OPNET and NS2. However, they suffer from drawbacks as it is difficult to build abstract models for quick studies of alternate scenarios and variation of system parameters. The interested reader can refer [18] for a comparison of these various simulators. DEVS also facilitates the development of Experimental Frames which help in easily configuring various scenarios and generating results that can help in comparing the performance of different system configurations. The objective of this research is to examine how DEVS can be used as an effective tool to model the system behavior at abstract levels and develop experiments to analyze the performance of various system configurations.

Specifically, in this study, the SCOPE communication system [2] was considered as a suitable system to study the effects of building abstract models of various components. Like any communication system, the SCOPE system too has many of its own protocols and hardware components. It is precisely because of its so many components and protocols that the system becomes fit for the study outlined above. The behaviors of many of these components are abstracted to a level required to study the performance of the whole system.

Although the component behaviors were available, the exact system configurations such as flight patterns or the aircraft, message generation patterns and so on were not available. Hence, the set of experiments conducted and results yielded are only indicative. They demonstrate the use of abstract models built using the DEVS formalism in studying the system performance and the effects of any change in the configuration, rather than provide exact recommendations to the designers. This study, therefore, is more a "proof of concept" research effort than a specific set of design recommendations. However, it should be noted that, if the exact details of the system configuration are provided, specific recommendations can be made by running various simulations using the models developed as part of this effort.

1.2. System Planning

Complexities in the behavior of any large scale system arise due to two factors: the dynamics of individual components and the structural relationships between components. Design decisions regarding either of these factors would have significant impact of the overall performance of the system under test (SUT). For example, let us consider a communication system involving modems and radios. Modems will have their own parameters such as error correction schemes, waveforms, speed of transmission, etc. The radios also will have their own parameters such as scanning rate (the rate at which channels are scanned), time taken to establish a communication link, etc. Each of these characteristics affects the system performance in some way or the other. Let the modem and the radio be connected now. This interaction itself will affect the system performance. New design questions arise. Should the modem directly interact with the radio? Or should there be a separate component which connects these two components?

The present research is an attempt to show how abstract models can be constructed using DEVS, which can then be used to study various design decisions taking into account both the individual dynamics and structural relationships.

1.3. Tools

Various tools were used in this study. In this section, some of the tools used in this research are briefly described.

1.3.1. Modeling and Simulation Formalisms

There are four fundamental modeling formalisms [1] for specifying dynamic system behaviors, namely, Differential Equation System Specification (DESS), Discrete Time System Specification (DTSS), Qualitative System Specification (QSS) and Discrete Event System Specification (DEVS). Figure 1 depicts the basics of these four methods.

In this research, the DEVS formalism is used since the time base of the SUT is discrete and the observation intervals of the components of the SUT individually are piecewise constant over time. The DEVS formalism is a powerful tool which can be used to specify the behavior of the various components of a system.

1.3.2. Java

The Java programming language has created a lot of excitement with its promise of providing portability of applications. Java, as such, provides three forms of portability, viz. source code portability, CPU architecture portability and OS/GUI portability. It is due to the packaging of the Java Programming Language, the Java Virtual Machine and the class libraries associated with the language. The programming language provides source code portability which is the most common form of portability.

In addition to this, the programming language uses the object oriented paradigm. It also provides a rich set of libraries. Using these features, it is possible to create applications using the Object Oriented Analysis and Design approach (OOAD). In this

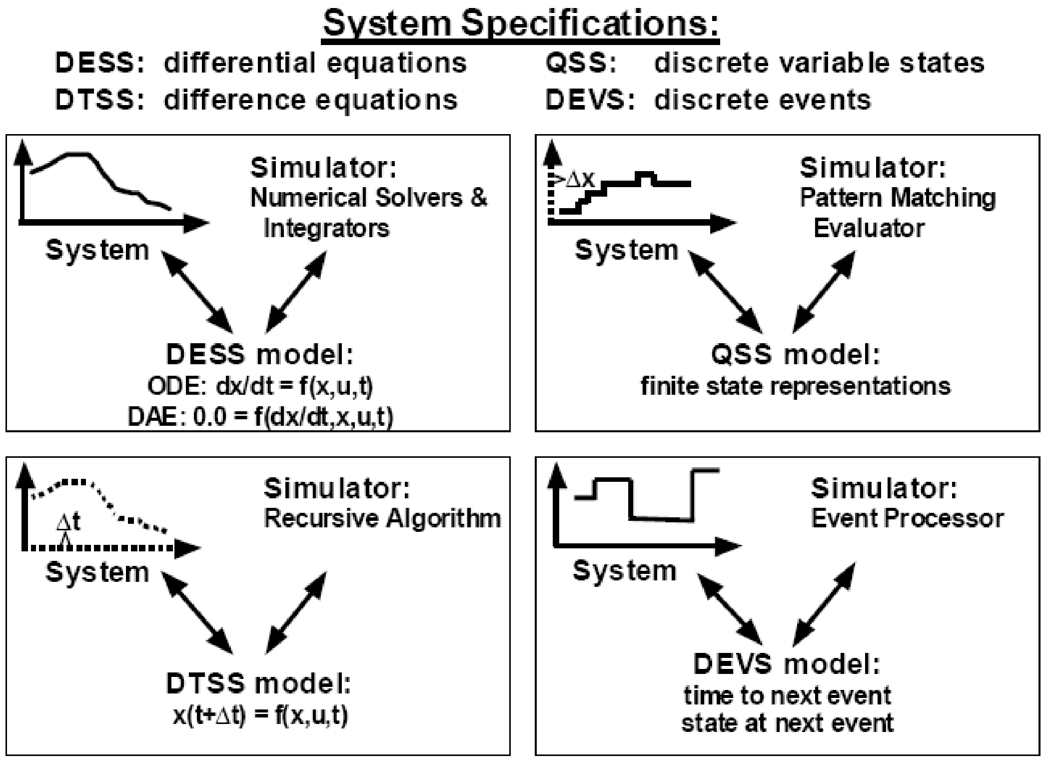

Figure 1. System Specification Formalisms

research we use the DEVS-Java environment which basically is a Java implementation of the DEVS formalism. Since many new objects need to be created which adhere to the DEVS formalism, Java is a perfect tool enabling such development work.

1.3.3 UML

Current software systems are very complex. Hence, there is a need for formally specifying the structure of such a system. The Unified Modeling Language has turned out to be a very useful tool in this regard. It is a graphics based language for specifying, constructing, documenting and visualizing software systems.

1.4. Plan of the Thesis

In the next chapter, the background information for the study and related work are described. Chapter Three focuses on the Experimental Frame concept and provides a theoretical introduction. Chapter Four explains the models that have been constructed as a part of this research. In Chapter Five, the various experiments that were conducted and the results obtained are documented. Chapter Six contains the conclusions of the thesis and indicates some proposals for future work.

2. Background

In this chapter, the SCOPE communication system that is modeled is explained and its components are described. Existing network simulators are briefly mentioned and compared. The DEVS formalism is also introduced. A DEVS based network model is also briefly studied to explore DEVS as a suitable tool to study network behavior.

2.1. SCOPE

SCOPE Command is a highly automated, high-frequency (HF) communication system that links US Air Force command and control (C2) functions with globally deployed strategic and tactical airborne platforms [2]. It is a highly reliable, cost effective system based on COTS/NDI equipment. This system is the primary command and control for military air forces and supports SITFAA, Mystic Star, DCS entry, National Command authority dissemination of emergency action messages and other missions requiring global HF Connectivity. It is designed to comply with the DOD Information Technology Security Certification and Accreditation Process (DITSCAP) for transmission of secure information. It provides high assurance data encryption, virus protection, intrusion detection, and user identification and authentication processes.

Figure 2. The SCOPE Command System

The system comprises of a worldwide network of fifteen HF ground stations that are remotely controlled from a Central Network Control Station (CNCS) located at Andrews AFB. This system provides HF voice and data communications to aircraft around the world in support of the following missions:

- Mystic Star
- USAF Global HF
- DCS HF Entry
- SITFAA

Each remote HF radio station is connected to the CNCS by DISA long haul circuits. CNCS operators can originate/answer HF radio calls and landline calls from/to the associated remote HF ground station. The system has the capability to select the best remote station and frequency to communicate with a specific aircraft, regardless of its location or propagation conditions.

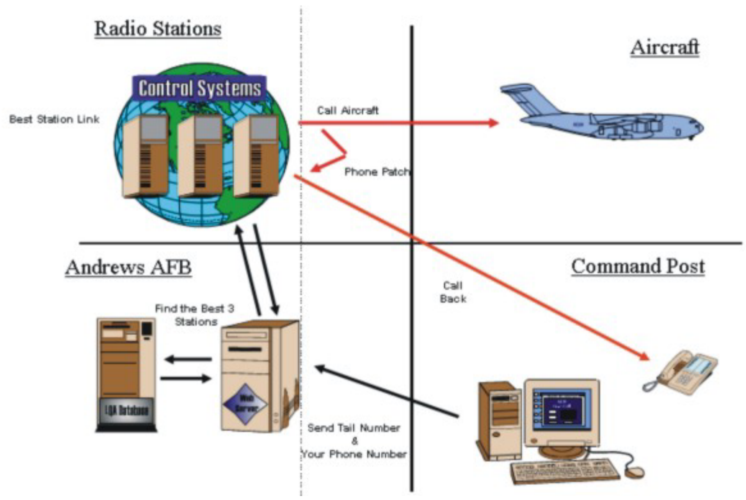

Figure 3. Modular Open Architecture of the SCOPE Command System

It increases overall operational and mission capabilities while reducing operation and maintenance costs. Its open architecture design permits flexibility to meet the changing mission and force requirements, centralized network control using "lights-out" (unmanned) station operation, and cost effective upgrades or network expansion.

The modular, open architecture system design helps to accommodate easily equipment upgrades and additions, or additional network sites. The ability to scale the network coupled with the SIL simulation and support capabilities provide a system design ideally suited for other worldwide applications requiring reliable, seamless HF communications.

Not only does SCOPE Command easily and economically expand to meet new HF mission requirements, but it fulfills next-generation mission roles as well. The open architecture and control software have a built-in flexibility to add next-generation capabilities such as multi-media and multi-band operations.

2.1.1. Typical System

A typical SCOPE Command station includes operator consoles, circuit switching equipment, HF radios, RF matrixes, and antennas. Station management and control use a standard PC workstation. It provides for full automatic operation and maintenance of local and remote site equipment as well as control of the local and wide-area networks (LAN/WAN).

A non-blocking digital electronic switch connects the station to the local military and/or commercial telecommunication services. The switch features unlimited conferencing, modular sizing, digital switch network, precedence function, and capacity for up to 2016 user lines.

The HF radio equipments include a DSP Receiver/Exciter. The radios feature Automatic Link Establishment (ALE) and Link Quality Analysis (LQA) capability and are adaptable to future ECCM waveforms. FSK, MIL-STD-188-110B and STANAG 5060 .Serial and 39 tone HF modem waveforms ensure backward-and-forward capability and mission interoperability. The transmit subsystem includes 4-kW solid-state power amplifiers, a high-power transmit matrix, and a combination receive/multicoupler antenna matrix.

SCOPE Command uses a modular, open-system design to automatically manage and control all network operations, including those at split-site stations. To achieve maximum flexibility, the system uses commercially available standards-based software and a multitasking operating system. This approach permits 14 out of the 15 network stations to operate "lights out" (unmanned) and to be economically controlled from a central location. The control system also includes LAN software, servers, and routers to support unlimited LAN/WAN.

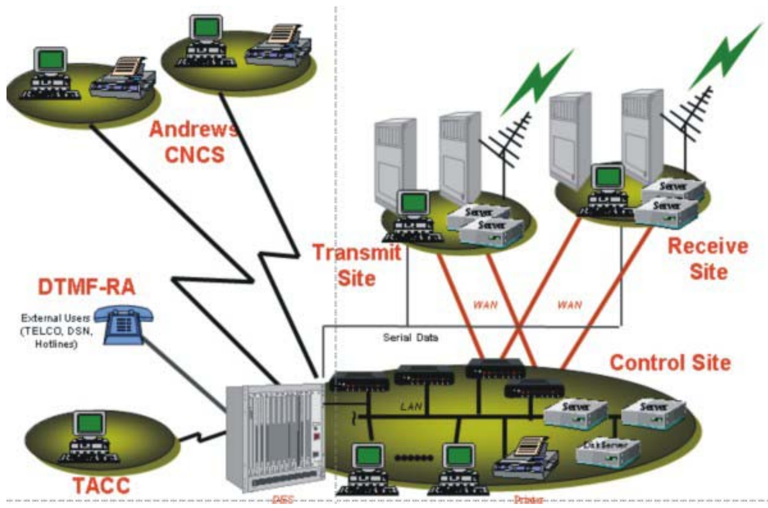

Figure 4. Typical System

2.2. HF Radio Waves

The ionospheric channel exhibits temporal effects over a wide range of time scales, including multipath spreads of the order of milliseconds that produce intersymbol interference, various types of fading of the order of seconds to minutes, hourly diurnal variations, etc up through the 11 year sunspot cycle [3]. However, technologies have been developed to deal with these challenges. Such is the value of beyond the line of sight wireless communications that HF radios are being used to carry Internet traffic.

A HF skywave channel conveys signals beyond the line of sight through refraction from the ionosphere and possible intermediate bounces off the earth's surface to one or more receivers. The refractive and absorptive characteristics of the ionosphere layers depend on the frequency of the HF radio signal. Hence, the primary requirement of an automated HF radio system is the identification of a usable frequency.

In current systems, data from various measurements and prediction techniques may be combined to select a frequency. A very common measure is the Link Quality Analysis (LQA) score which is related to the signal to noise ratio for that particular frequency [4]. It should be noted that in the following discussions, the terms channel and frequency may be used interchangeably. A channel corresponds to a frequency, e.g., channel 1 may imply the use of frequency 2 MHz and so on. Once a frequency is chosen, an Automatic Link Establishment (ALE) protocol oversees the coordination of the different participating radios to that particular frequency and the transition to data transfer.

The audio channel provided by a HF radio usually has a bandwidth of 3 KHz and exhibits a low and fluctuating signal to noise ratio. This is bounded by limits on radiated power and by galactic, atmospheric and man-made noises. Signals reach the receiver via refractions through one or more ionospheric layers, each of which may be in motion. The received signal is thus a composition of multiple signals having independent time-varying path losses and phase shifts. Multipath interference, deep fades and impulsive noise affect the SNR which may result in the ALE having to select a new frequency.

2.3. Automatic Link Establishment

The traditional method of operating HF communication systems involves manual time-frequency planning and coordination. It also needs skilled radio operators to perform frequency selection, frequency monitoring and link establishment. By using an Automatic Link Establishment protocol, selective calling, preset channel scanning and real time channel propagation evaluation can all be automated thus achieving automatic connectivity.

The basis for ALE frequency selection is the Link Quality Analysis (LQA) [5]. The LQA is determined by analyzing the signal characteristics, signal to noise ratio and delay distortion, of the data signal used in sounding or initiation of an ALE call. A LQA is performed continuously on the channels within the ALE scan list. A database of LQA values is generated and updates continuously.

The current ALE protocol being used is of the third generation. The third generation ALE protocol is summarized in MIL-STD-188-141B Appendix C [6]. Second generation HF automation was used earlier to provide a sufficiently robust, reliable and interoperable ALE technology for using HF radio in long-haul and mobile networks. This was later extended to provide data applications over HF with the addition of a robust data link protocol. As the HF networks grew, it was quickly realized that overhead traffic needed to be reduced. One of the primary goals of the third generation HF automation was to support data traffic bursts in peer to peer networks with hundreds of stations. The resulting limits on linking, message delivery and routing table maintenance improved performance in the smaller networks as well.

However, in the current SCOPE Command system, the second generation ALE is the one that is used. Accordingly, the following sections explain the basics of this protocol.

2.3.1. Second Generation ALE

The ALE techniques and detailed operation are specified in the MIL-STD-188-141B Appendix A [5].

2.3.1.1. General Requirements for ALE operation:

Addresses: There is a specific addressing scheme for ALE operation. A digital addressing structure based on a standard 24 bit (or 3 character) word and the basic 38 character subset is used. The basic 38 set includes all capital alphabets (A-Z), all digits (0-9) and designated wildcard and utility symbols. The addressing scheme allows a station to address another individual station, multiple stations or special modes.

Scanning: The radio system is capable of scanning selected channels continuously under either manual or automated control. Scanning stops when the radio receives a manual instruction to do so or receives an input from the controller or if its external stop scan line is activated by any means.

Calling: On request, the radio should be capable of placing a call according to the calling protocol explained later in section 2.3.1.5.

Channel Evaluation: The radio system also automatically transmits sounding signals and measures the signal quality of ALE receptions.

Channel Quality Display: If an operator display is provided, the display has a uniform scale of 0-30. This is based on the signal plus noise plus distortion to noise plus distortion (SINAD) measurement.

2.3.1.2. Operational Rules:

Table 1 lists the operational rules for the ALE operation.

1)	Independent ALE receive capability (in parallel with other modems and simular audio receivers) (critical).
2)	Always listening (for ALE signals) (critical).
3)	Always will respond (unless deliberately inhibited).
4)	Always scanning (if not otherwise in use).
5)	Will not interfere with active channel carrying detectable traffic in accordance with table A-I (unless this listen call function is overriden by the operator or other controller).
6)	Always will exchange LQA with other stations when requested (unless inhibited), and always measures the signal quality of others.
7)	Will respond in the appropriate time slot to calls requiring slotted responses.
8)	Always seek (unless inhibited) and maintain track of their connectivities with others.
9)	Linking ALE stations employ highest mutual level of capability.
10)	Minimize transmit and receive time on channel.
11)	Automatically minimize power used (if capable).
NOTE : Listed in order of precedence.	

Table 1. Operational Rules of ALE

2.3.1.3. Operational Overview:

The fundamental protocol exchange for link establishment is a three way handshake. An ALE controller may be in one of the three conceptual states as shown in figure below. In the Available state, the ALE is either scanning or sounding and is available to either make a call or receive one. In the Linking state, the ALE is in the process of establishing a link with another ALE. The ALE in question may have either initiated or received the call. In the Linked state, the ALE has established a link with another ALE and is not available for use by any other entity.

A station which is trying to establish a link sends ALE calls on the scanned channels in an order dictated by a channel selection algorithm. It links on the first channel that supports a link with the called station. If a channel is rejected after linking, the ALE controller terminates the link and updates the LQA data with measurements obtained during linking. During the scanning-calling cycle, if a caller comes across busy channels it (the caller) skips the channels to avoid causing interference.

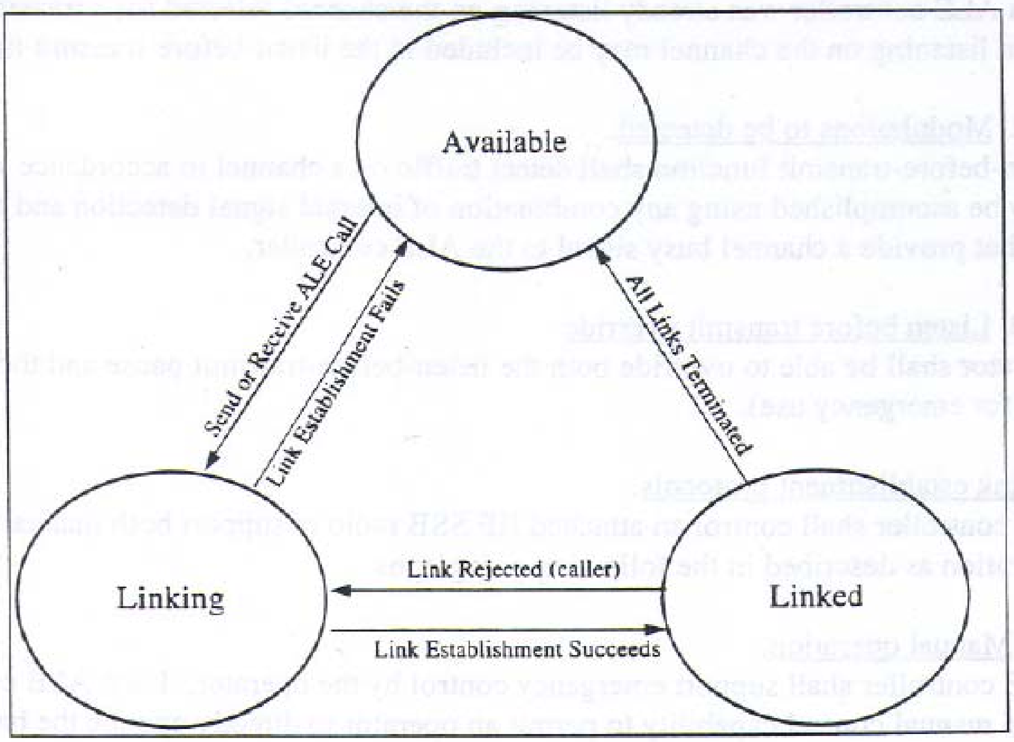

Figure 5. ALE Conceptual States

After all the available channels have been tried and if linking is still unsuccessful, the caller can come back to these channels and attempt to call on them if they are free. If a calling station has exhausted all of its preset scan channels, it returns to the normal scanning state or the available state. Normal scanning state is also the receiving scanning state. It also alerts the operator or controller about the unsuccessful attempt. There are also specific end of frame detection methods, in which ALE controllers search for specific words that signal the conclusion of the ALE signal. There are only certain words that are valid conclusions. For more details, [5] may be referred to.

2.3.1.5. One – to – One calling:

The protocol to establish a link between two individual stations consists of three ALE frames: call, response and acknowledgement. The sequence of these events and the timeouts involved which constitute the three way handshake are described here briefly using a calling station, SAM, and a receiving station, JOE. For a more detailed specification, please refer to [5].

Sending an individual call:

After selecting a channel for calling, the calling station, SAM, begins the protocol by first listening on the channel to avoid disturbing any active transmissions on it. It then tunes to that channel. If the receiving station, JOE, is known to be listening on the channel and not scanning, SAM transmits a single channel call that contains only a leading call and a conclusion. Otherwise, it sends a longer calling cycle that precedes the leading call with a scanning call of sufficient length to capture JOE's receiver.

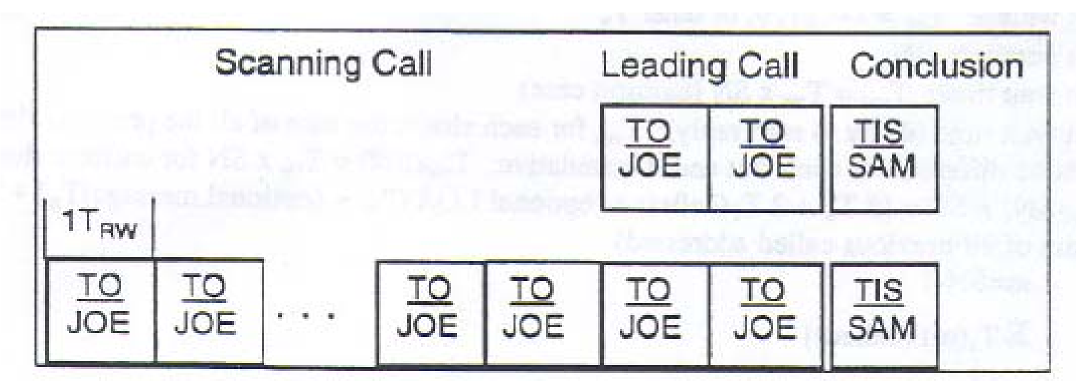

Figure 6. ALE Individual Calling

The calling station then waits for a preset reply time, Twr, to receive the called station's response. If the expected reply is not received within this time, then the linking attempt has failed. At this point, if there are other channels that have not been tried, then the linking attempt will start over a new channel. Otherwise, the ALE controller returns to the available state and informs the calling station about the unsuccessful attempt.

<u>Receiving an individual call:</u>

When the called station, JOE, arrives on a channel during its scanning cycle, it tries to detect any available ALE signal within its dwell time. If an ALE signal is detected and word sync is obtained, it examines the received word to determine the appropriate action.

If the call is addressed to itself, then the ALE controller stops scanning and enters the linking state. It continues to read all the ALE words while waiting for a preset time (denoted by Twce) for the calling cycle to end and message conclusion to begin.

- If the received word is potentially from a sound or some other protocol, the controller processes the word accordingly.
- Otherwise, it resumes its previous state

While in the linking state, the controller evaluates each received word. It immediately aborts the handshake and returns to the previous state when any of the following events occur:

- It does not receive the start of a quick ID, message or frame conclusion within Twce or the start of a conclusion within a certain time period, Tmmax, after the start of message section.
- It receives any invalid sequence of ALE words, except during the reception of a scanning call when it tolerates up to three contiguous words containing uncorrectable errors.

- It does not detect the end of a conclusion within a time period, Tlww, after the first word of the conclusion.

If it receives the start of a quick-ID or a message section within Twce, it attempts to read one or more complete messages within a new preset time Tmmax. If a frame conclusion starts, it waits to read the calling station's address within a preset time Txmax. If an acceptable conclusion sequence is read, it starts a last word timeout Tlww = Trw while searching for additional words and the end of a frame. Its response is then triggered. However, if the conclusion signaled the end of the linking attempt, it does not respond and will return to its previous state.

If all of the above conditions are satisfied, the called station initiates an ALE response immediately after detecting the call unless otherwise directed by the operator or controller.

Response:

If there is no other traffic on that channel, the called station tunes up and sends a response (accepting the call) and then starts its own reply timer Twr. If the channel is in use, the ALE controller ignores the call and resumes its previous state.

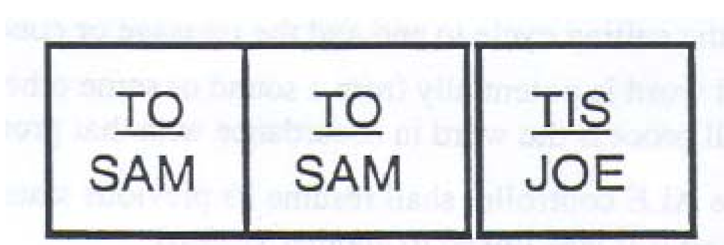

Figure 7. ALE Response

Receiving a response:

If the calling station reads a response successfully before the end of its Twr timer, it processes the rest of the frame in the same way that the call was processed by the called station. Specifically, the calling station terminates the linking attempt when:

- It does not receive an appropriate response within its Twr period.
- It receives an invalid sequence of ALE words.
- It does not receive an appropriate conclusion within Tlc.
- It does not detect the end of the conclusion within Tlww.

If the handshake is aborted for any reason, the calling station normally restarts the calling protocol on a different channel. If however, everything goes well, it sends an acknowledgement. If the response received signaled the end of the linking process, then the calling station aborts the process and informs the operator or controller accordingly.

Acknowledgement:

If an acceptable response is received, unless otherwise dictated by the operator or controller, the calling station alerts the operator or controller and sends an acknowledgement. It then enters into a linked state with the called station. It starts a wait for activity timer Twa. This causes the link to be dropped if there is no activity within that time, thus preventing extended periods of non usage.

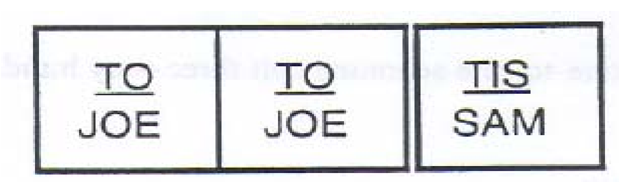

Figure 8. ALE Acknowledgement

Receiving an acknowledgement:

If the called station successfully reads the beginning of an acknowledgement within its Twr timeout, it processes the rest of the frame in the same way as described earlier. It either receives the end of the conclusion or aborts the handshake. Specifically, it aborts the handshake if any of the following occurs:

- It does not receive an appropriate acknowledgement within its Twr period.
- It receives an invalid ALE word sequence.
- It does not receive the start of conclusion within Tlc after the start of the frame.
- It does not detect the end of the conclusion within Tlww.

If it aborts the handshake for any reason, it returns to its previous (pre linking) state and informs its operator or controller about the unsuccessful attempt. If everything goes well, it enters the linked state with the calling station and starts its wait for activity Twa timer. It also alerts its operator or controller.

A typical one-to-one scanning calling three way handshake takes 9 to 14 seconds. For specific calculations of the timers mentioned here, please refer (reference number of MIL STD).

Link Termination:

Termination of a link after a successful handshake is accomplished by sending a frame concluding with a specific message to the linked stations whose links are to be terminated.

Termination can be done either manually or automatically. Automatic termination is achieved with the use of the wait for activity timers. These may be overridden manually.

Collision detection:

Due to factors such as interference, fading, etc, it is possible that the continuity of a received signal is lost. The ALE controllers use the Golay error correction and detection scheme. When one or both Golay words of a received ALE word contain uncorrectable errors, the controller attempts to regain word sync, with a bias for words that arrive with the same word phase as the interrupted frame. If word sync is reacquired at a new word phase, it is implied that a collision has occurred. The interrupted frame is discarded and the interrupting signal is processed as a new ALE frame.

2.3.1.6. Other calling schemes:

There are other calling schemes that are supported apart from the one-to-one calling. These are:

- One-to-many calling
- Allcall
- Anycall protocol
- Wildcard calling protocol

2.3.1.7. Sounding:

Sounding constitutes an important part of the ALE operation, although it is not part of the three way handshake protocol. The sounding signal is a unilateral one way transmission performed at periodic intervals on unoccupied channels. To implement sounding, there is a separate timer which initiates the controller to send sounding signals. Any station can receive sounding signals. As a minimum, the signal information is displayed to the operator and in stations equipped with connectivity and LQA memories, the information is stored and used later for linking. If a station has had recent transmissions on any channels that are to be sounded, it is not necessary to sound again on them until the sounding interval started since those transmissions have expired. In addition, if a set of stations is polled, their responses serve as sounding signals for the other receiving stations in the net. All stations are capable of performing periodic sounding on clear prearranged channels. The sounding capability and sounding interval are adjustable by the operator or controller.

The structure of the sound is virtually identical to that of the basic call. However, the calling cycle is not needed and there is no message section. It is only necessary to send the termination that identifies the transmitting station. There are both single channel and multiple channel protocols. In this report, only the single channel protocol is examined.

Single Channel:

The basic protocol consists of only one part, viz., the sound. The sound contains the transmitting station's address e.g., A. If A is encouraging calls and does indeed receive one, it follows up with an optional handshake protocol. This protocol is similar to the previous scanning call protocol, except that it is triggered by the acquisition of connectivity from the station that is to be called.

Consider two stations A and B, such that, A is scanning sounding and B is receive scanning and requires contact with A if heard. A uses the standard call acceptance scanning sound in which case B calls A. When ALE stations are scanning sounding and receptive to calls, this handshake must be used. The calling station immediately initiates the call upon determining that the station to be called has terminated its transmission. No wait time before transmit time is required. Hence, if B hears A's sound, it immediately calls A using the single channel call. Also if B's controller or operator identifies A, it can attempt the optional handshake.

If A plans to ignore calls, it advises the other stations accordingly and immediately returns to normal available state.

2.4. Link Quality Analysis

The basis for frequency or channel selection for transmitting ALE calls is the Link Quality Analysis (LQA) score. LQA is performed continuously on the channels in the ALE scan list. The ALE controller assigns an LQA score to each channel the radio scans by analyzing the signal characteristics (SNR) of the data signal of a remote station used in sounding or initiating an ALE call. A database of LQA scores is generated and maintained. These entries are continuously updated.

2.5. HF E-Mail System

The HF E-Mail system adds conventional e-mail capabilities to the SCOPE Command system [7]. Using this system, an operator in any aircraft can send an e-mail through a ground station to the central station. E-mails can be sent in the other direction too, that is, from the central station to any aircraft. Currently the system does not support sending messages from one aircraft to another. Figure 8 shows a simplified depiction of the system.

An interface is also provided to external networks. External networks are classified into two types: SIPRNET and NIPRNET. SIPRNET messages are contained in a secure network whereas NIPRNET messages are not. These are two parallel but separate networks. If a message is addressed to an aircraft or a remote subscriber, the HF E-mail servers obtain routing information from the LQA database. This database contains information regarding the best way to communicate with the remote subscriber, which in most cases, would be the best ground station and channel to use. The message is then delivered to the subscriber using the ALE radios at the ground station.

2.5.1. Functional Description

Two very similar HF E-mail networks are established at the CNCS, one for NIPRNET services and another for SIPRNET services. Figure 9 depicts the block diagram of the system. The basic functional components or services are briefly mentioned in the following sections.

2.5.1.1. External Network Interface:

The external networks are connected using routers. The SIPRNET router is connected to a nearby router that is part of the Andrews AFB SIPRNET WAN through an Ethernet connection. Similarly, the NIPRNET router is connected to a nearby router that is part of the Andrews AFB NIPRNET WAN through another Ethernet connection. The Network Manager is used to configure and manage the Ethernet switches and routers.

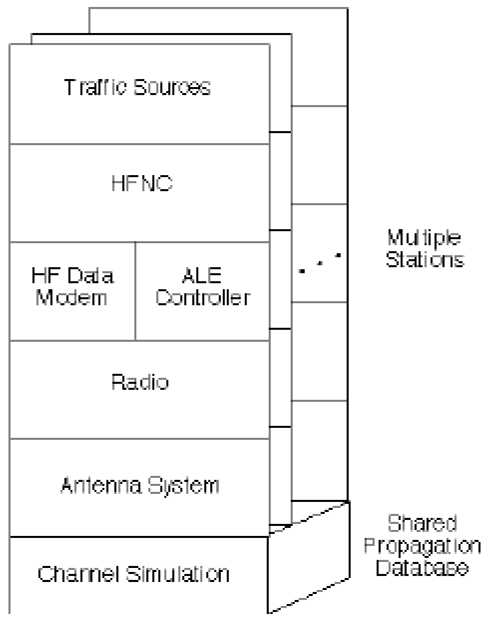

Figure 9. HF E-Mail

2.5.1.2. LQA Database:

The LQA database contains information regarding the best remote station to be used for communicating with a specific remote subscriber. This information is continuously updated as the conditions change due to aircraft movement, atmospheric conditions and other factors affecting propagation of HF signals.

2.5.1.3. SCOPE Command Stations Interface:

This interface consists of a HF Messenger (HFM), a crypto unit and a HF Modem. One HF Messenger computer is provided in each network for each remote station. These computers are locally connected using switches and connected to the remote stations using long haul circuits. In the SIPRNET network, the e-mail messages are encrypted using an encryption unit.

The HFM software running in the HFM computer is responsible for delivering e-mail messages to a remote subscriber using the Q9600 HF modem and the radio at each station. It controls the modems and radios. In the SIPRNET network, the radio and modem control lines are protected by optical isolators. The long haul circuit is actually a 9600 bps data circuit. It allows the HF Messenger to issue ALE control commands and receive ALE status from the RT-2200 receiver. The station assets are set into operation as an ALE group by the station operator. LQA reports from the receiver are forwarded to the LQA server at the CNCS for use in the best station selection when messages are to be delivered to a remote subscriber.

When the HF Messenger software receives a message to be delivered, it commands the receiver to initiate an ALE call with the message recipient. It also sets an initial data rate in the modem using the LQA scores received during the linking process. The data rate is set to match the channel conditions. Channels more conducive to error free communication would enable the use of higher data rates. After the ALE link is established, the HF Messenger delivers the message to the remote node using the protocols defined in the STANAG 5066. These protocols ensure error free delivery of messages. The data rate may be changed during message delivery based on the success rate of previous transmissions.

2.5.1.4. E-Mail Server:

The E-mail server is a conventional e-mail server running the Rockliffe MailSite software. This acts as an SMTP mail relay for the HF E-mail network. It initially receives all incoming messages from the NIPRNET or SIPRNET. If the message is addressed to the SCOPE Command domain name, it is sent for further processing and routing to the DAER. The server also receives messages from the DAER to be routed to external networks.

2.5.1.5. Network Manager:

The Network Manager provides typical network monitoring and management capabilities for the network.

2.5.1.6. Domain Controller:

Each network contains two identical domain controllers. These provide the following key functions:

- Domain login services required by the E-mail and DAER/GRM servers
- Active directory services including user management
- Centralized management of security policy
- Internal DNS
- Centralized time source

2.6. HF Email Delivery

2.6.1. Ground to Air

The messages sent by external networks are first received by the Email Server using standard SMTP. The message is then forwarded to the DAER. The DAER sends a query to the GRM, requesting a list of the best stations to use for forwarding the message to the aircraft. Upon receiving this list, the DAER communicates with the HF Messenger software in each of the best stations to find their current status and availability.

- If a station is in an ALE call with the required aircraft or is attempting one, that station is chosen.
- If no station is in a link, the DAER chooses the best station that is not busy (linked with some other ALE).
- If all stations are busy, the DAER continuously polls the stations until one of them becomes available.

The DAER uses SMTP to forward the message to the stations.

In the station, now, the HF Messenger software initiates an ALE call with the aircraft. Once the link is established, the software transmits the message to the HF Messenger software in the computer onboard the aircraft. If the station is not able to forward the message, it returns the message to the DAER, wherein the DAER treats this as a new request and repeats the procedure.

When the DAER does not receive a best station list from the GRM because the aircraft data is not available, the DAER places the message in a hold queue for later delivery.

2.6.2. Air to Ground

Email transmission from an aircraft to external NIPRNET/SIPRNET users occurs in two ways.

The first method is essentially the reverse process of the ground to air transmission. The ALE in the aircraft initiates an ALE call with a SCOPE Command ground station. After the link is established, the HF Messenger in the ground station receives the email and forwards the message to the DAER. If the message is addressed to the SCOPE Command domain, the DAER then initiates delivery to the remote subscriber. Otherwise, it sends the message to the server which, in turn, delivers the message into the NIPRNET/SIPRNET.

The second method is used in cases where users do not spend too much time in scanning ALE channels and recording best station data and so on. In such cases, the aircraft is allowed to sound in such a way that the SCOPE Command network will then call the aircraft using the sounding data to select the best station. Once a link is established, the aircraft can deliver the email to the HF Messenger at the ground station.

2.7. Objectives of this research

With this SCOPE HF E-mail system in place, the network designers have some typical questions about the performance of the system [4]. Some of these questions are:

- How much traffic can the current system handle?
- Would increasing the available frequency set improve the system?
- Would increasing the number of levels at each station improve the system?
- How important is it to have stations within the theater of operations?
- What are the effects of adding additional ground stations?
- How do good/poor propagation conditions affect the system?
- How could the current system be improved by changing/modifying operating parameters?

The motivation for this research arises from the need to answer these questions. Although extremely detailed high resolution models which follow every word in the ALE protocol can be built and the working of the ALE radios simulated, it is doubtful how useful they would actually be in answering high level questions regarding the system performance such as the ones mentioned above. Hence, this research investigates the building of abstract (or low detail or lumped) models of the ALE radios and HF stations using the Discrete Event System Specification (DEVS) formalism and using these models to simulate a system under various conditions and parameters to answer some of these questions. This research also investigates the development of an Experimental Frame which can run experiments on the different system designs and analyze their performance.

One particular advantage of using DEVS needs to be highlighted in this particular context. It is possible with this approach to build models at the appropriate level of detail and tailor them to specific needs. It also helps that this development can be done rather quickly. DEVS further, provides the concept of Experimental Frame (EF). It is this EF that can generate the various experiments that can be simulated using the abstract models mentioned above. Thus there is a clear distinction between the actual system in question and the experiments that are run on this system.

In building these models and experimental frames, it is important that existing high detail simulators, if any, are not ignored or passed over. Rather they could be examined and the useful or required features from these simulators could be used in the DEVS models to prevent the reinvention of the wheel. Accordingly, in this research, the

NETSIM simulator is examined and many of its features are used in building the DEVS models, especially for the ALE and HF channels. The next few sections introduce some of these concepts.

2.8. NETSIM

NetSim is a discrete event simulator designed and implemented by Professor Eric Johnson. It was developed to support the systems engineering of high-power HF radio networks and has been independently validated by the US Defense Information Systems Agency (DISA) Joint Interoperability Test Command (JITC). It can be used in the simulation of large networks of fixed and mobile radios.

NetSim is actually a family of simulators. The NetSim family of simulators implements a discrete-event communications network simulation architecture depicted in figure 10 [8]. Each of the modules shown functions independently and implements its respective function at a level of detail appropriate for the investigation.

• Traffic sources generate voice or data messages according to specified inter-arrival time and message size distributions.

• The HF Node Controller (HFNC) at each station implements network-layer protocols and station-wide control.

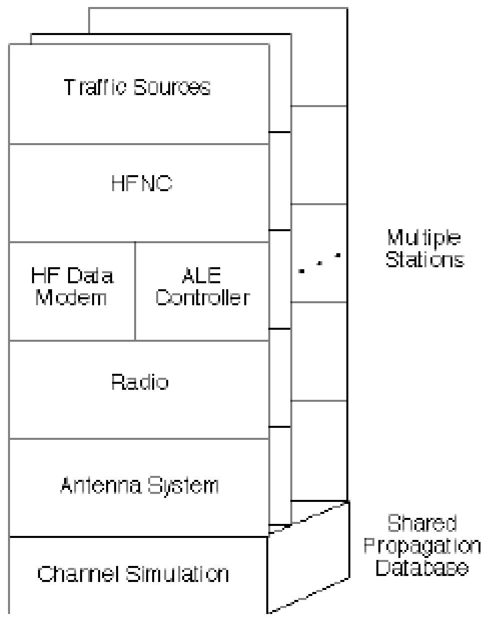

Figure 10. NetSim Family of Simulators

• ALE controllers implement the ALE protocol and waveform under study. (The waveform is simulated as the probability of correct frame reception vs. signal-to-noise ratio).

• Radio and antenna modules determine power and noise levels, intermodulation distortion, gain versus azimuth, etc.

NetSim implements the ALE protocol in some detail. It mimics the actual ALE operation to a large extent. At the same time, however, there is no implementation of the three way handshake and exchanging of actual messages. It basically checks whether a connection can be established from a given aircraft to any of the ground stations on any channel. Another major feature of NETSIM is its ability to approximately predict path losses between pairs of stations while minimizing both memory used and CPU utilization during a simulation.

2.8.1. SNR Prediction

In a HF radio simulation, the model used for the channel is very important. It is critical to model the dependence of link propagation on the time of day, sunspot activity, the frequency chosen, the types, locations, and orientations of the sending and receiving antennas, and the other equipment and protocols in use. The overall performance of an HF network is the result of how effectively it uses the links and stations available. Due to the complexities of skywave propagation, HF radio networks are less amenable to analytical modeling than other networking technologies, and usually require simulation for accurate performance prediction [9].

Where ALE is concerned, the performance over these channels is almost entirely a function of the SNR. The SNR on a HF link is a function of the following variables:

- Transmit antenna location
- Receive antenna location, and gain in the direction of propagation
- Transmit effective radiated power
- Receiver effective noise
- Frequency
- Time of day
- Day of year
- Sunspot number

Useful SNR predictions for each link can be derived from IONCAP or one of its descendants (e.g., VOACAP or ICEPAC). Such prediction programs use a combination of measured data and generally-respected algorithms to make statistical predictions of signal strength for arbitrary links. When combined with noise estimates, the programs produce estimates of median SNR over a link as well as estimates of the first and ninth deciles of SNR. These can be used to generate representative random processes for the SNR values on the links of interest.

When a radio is tuned to a particular frequency, it receives a composite signal that includes the effects of all transmissions worldwide that are in progress on that frequency. To compute the effective SNR of the composite signal (for receivers that lock onto the strongest arriving signal), it is sufficient to compute the ratio of the strength of the strongest arriving signal to the sum of natural noise plus all other signals plus distortion. The channel model used in most NetSim simulations is the “Walnut Street” model of ionospheric propagation [10], along with a direct wave model for aircraft within line of sight at altitude. Estimates of median SNR and first and ninth deciles of SNR for a link are computed using VOACAP [19]. These are then used to generate representative random processes for path loss on the links of interest, using linear interpolation between values predicted for every hour plus lognormal variation about this line for intermediate-term variation plus Rayleigh fading. For comparison with other simulations, however, the Walnut Street model can be replaced with a fixed-SNR model.

2.8.2. Channel Simulation in Static Networks

If all stations are stationary, the SNR deciles for all links are computed once and reused throughout the simulation. Denoting the hour by subscript h, the frequency by subscript f, the sending station by subscript s, and the receiving station by subscript r, the first, fifth, and ninth deciles of the SNR, $S1hfsr$, $S5hfsr$, and $S9hfsr$ can be precomputed using a prediction program. When the SNR on a link is required during simulation, the appropriate set of three channel parameters is retrieved, two uniformly distributed random numbers $u1$ and $u2$ are drawn, and an estimate of *SNRhfsr* is computed as follows. First, a normally distributed random variable z is computed from $u1$ and $u2$:

$$z = cos\ 2\pi\ u_1\ \text{-}2\ ln\ u_2$$

Then the sign of z is used to determine whether the SNR value will be above or below the median:

$$SNR_{hfsr} = \begin{cases} S_{5hfsr} + z\left(\dfrac{S_{5hfsr} - S_{1hfsr}}{1.28}\right) & z < 0 \\ S_{5hfsr} + z\left(\dfrac{S_{9hfsr} - S_{5hfsr}}{1.28}\right) & z \geq 0 \end{cases}$$

2.8.3. Channel Simulation in Dynamic Networks

In the case of mobile stations, the method outlined above leads to a huge number of calculations and accesses to the channel simulation routine. So, a new interpolated propagation method is used.

Tables containing the values of parameters for each link between a fixed station and grid points located through out the world can be precomputed. These tables are very useful in mixed static/dynamic networks. When the SNR is required for any link that involves one of the fixed stations, the parameters from the corresponding table for the corners of the grid square that encloses the other station (fixed or mobile) are retrieved and interpolated among those sets of parameters to estimate the parameters for the interior point.

For example, an arbitrary propagation parameter X would be computed as follows for the position of the aircraft shown in figure 11:

1. Retrieve parameter values XSW, XSE, XNW, and XNE from the table of precomputed values for the fixed station at the other end of the link.

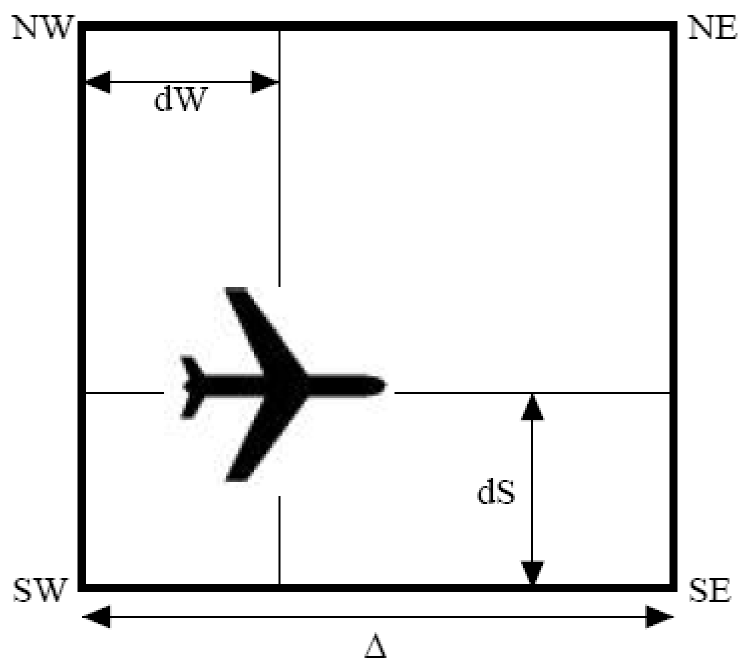

Figure 11. Projection of the SNR based on position

2. $$X = X_{\mathrm{SW}}\left(1-\frac{dS}{\Delta}\right)\left(1-\frac{dW}{\Delta}\right) + X_{\mathrm{SE}}\left(1-\frac{dS}{\Delta}\right)\left(\frac{dW}{\Delta}\right) + X_{\mathrm{NW}}\left(\frac{dS}{\Delta}\right)\left(1-\frac{dW}{\Delta}\right) + X_{\mathrm{NE}}\left(\frac{dS}{\Delta}\right)\left(\frac{dW}{\Delta}\right)$$

As mentioned earlier, these fast propagation predictions of NETSIM can be utilized by the DEVS models.

2.9. ICEPAC

ICEPAC stands for Ionospheric Communications Enhanced Profile Analysis and Circuit [19]. It is a propagation predictions model whose predictions are normally used for long term frequency management and circuit planning. They are also often used for hour to hour and day to day operations. The ICEPAC technical reference manual provides a lot of insight into the actual prediction methods. In this study, the ICEPAC data was used to predict the SNR between a mobile station and a ground station at a given time of day, the month, year and sunspot cycle.

3. SCOPE Command Simulation Building Blocks

In this chapter, the DEVS formalism is explained briefly and the concept of the Experimental Frame is further elaborated upon. To examine the applicability of DEVS as a tool to observe network performance, a very simple network containing nodes and links is modeled and some simulations are run.

3.1. DEVS

Originally introduced as a formalism for discrete event modeling and simulation, the DEVS (Discrete Event System Specification) methodology has become an engine for advances within the wider area of information technology. A variety of systems theory and artificial-intelligence applications employing DEVS-based concepts have been developed. Discrete event processing is characterized by the ability to perceive the flow of sensory stimuli as discrete events, and to attend to both sequencing and timing of such events. The Discrete Event System Specification formalism provides a means of specifying a mathematical object called a system. Basically, a system has a time base, inputs, states, outputs, and functions for determining next states and outputs given current states and inputs. Discrete event systems represent certain constellations of such parameters just as continuous systems do.

A Discrete Event System Specification (DEVS) is a structure [11]

$\boldsymbol{M} = \langle \boldsymbol{X}, \boldsymbol{S}, \boldsymbol{Y}, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, \ \boldsymbol{ta} \rangle$ where

$\boldsymbol{X}$ is the set of input values
$\boldsymbol{S}$ is a set of states,
$\boldsymbol{Y}$ is the set of output values

δ_{int}: S x S is the *internal transition* function
δ_{ext}: Q x X_b -> S is the *external transition* function, where
$\boldsymbol{Q} = \{(s, e) \mid s \, \varepsilon \, S, 0 \leq e \leq ta(s)\}$ is the *total state* set
e is the *time elapsed* since last transition
$\boldsymbol{X_b}$ denotes the collection of bags over X (sets in which some elements may occur more than once).
δ_{con}: Q x X_b -> S is the *confluent transition* function,
λ: S -> Y_b is the output function
***ta*:** S –> $R^{+}_{0,\infty}$ is the *time advance* function

The structure denoted above represents a *parallel DEVS* model. It follows the Parallel DEVS formalism in that it can handle simultaneous inputs. DEVS models can also be *sequential.* DEVS models can be classified into other major types: *atomic models* and *coupled models*. A DEVS atomic model specification defines the states (variable values) and associated time bases resulting in piecewise constant trajectories over variable periods of time. The atomic model specification also defines how to generate new state values and when new states should take effect.

A DEVS coupled model designates how atomic models can be coupled together and how they interact with each other to offer more complex functionalities. Given atomic models, DEVS coupled models are formed in a straightforward manner. Two major activities involved in *coupled* models are specifying its component models, and defining the coupling that represents desired interactions.

A Parallel DEVS *coupled* model [1] is formally defined by:

DN = < X, Y, D, {*Mi*}, {*Ii*}, {*Zi,j*} > where

X is a set of input values
Y is a set of output values
D is a set of the DEVS component names.
For each i ε D,
Mi is a DEVS component model

Ii is the set of influencees for I.
For each $j \varepsilon Ii$,
Zi,j is the i-to-j output translation function.

Yet another DEVS structure is the Variable Structure DEVS which has the capability of dynamic reconfiguration [13]. The interested reader may refer [13] for more details.

3.2. Experimental Frame

An Experimental Frame (EF) is a specification of the conditions under which a system is observed or experimented with [12]. The experimental frame is the operational formulation of the objectives motivating a modeling and simulation project. Many EFs can be formulated for the same system and the same EF can be applied to different systems. This is a result of the fact that there might be different objectives in modeling a system or that different systems can be modeled with the same objectives.

There are two views of an Experimental Frame. The first holds that the EF is a definition of the type of data elements that go into the database. The second views the EF as a system that interacts with the system of interest or system under test to obtain data under specified conditions. In this second approach, a typical experimental frame, as depicted in figure 12, consists of a *generator*, an *acceptor*, and a *transducer*. The *generator* stimulates the system under investigation in a known, desired fashion. The *acceptor* monitors an experiment to see that desired conditions are met. The *transducer* observes and analyzes the system outputs. The experimental frame concept provides a structure to specifying the simulation conditions to be observed for analysis.

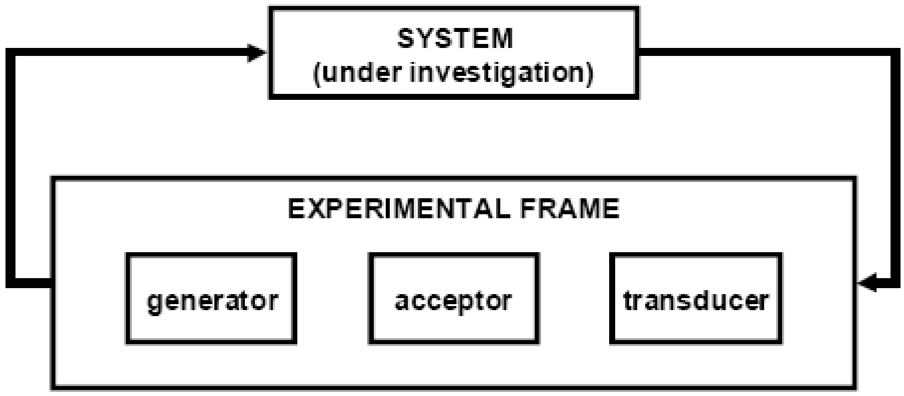

Figure 12. Typical Experimental Frame

The design of an EF for a particular system is objective driven. Design objectives are first transformed into performance indices. An EF is then designed to simulate the system and measure the desired performance indices. When experimental frames are designed following this methodology, different systems modeled with the same objectives can end up having the same EF that tests them. Figure 13 illustrates such an event.

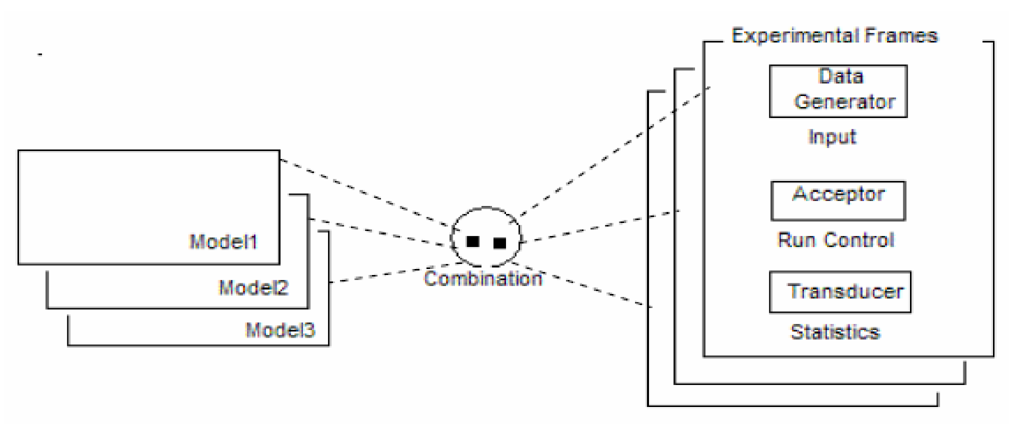

Figure 13. Objective Driven Experimental Frame

A more rigorous and mathematical approach to Experimental Frames and their design may be found in [20].

3.3. A Simple Network Model

The objective of this simple network model is to give an insight into how models can be quickly built at a higher level. These models are tailored to a given situation, but at the same time could also be modified to have a more general behavior. The ultimate goal of modeling the components and packets in the system would be to find out if the links in the network are overloaded or not. In order to measure this, it is enough if the packet size is known. It is not necessary to set up the application, transport and network layer protocols for packet transmission. All these protocol details could be abstracted simply by placing enough bytes in the network. In fact, in actual terms of network load, this is what the protocols essentially add, more bytes in the system. Since the network is ideal and errors and packet losses are not considered, these abstractions do not remove any functionality from the system model.

The current implementation supports the modeling and simulation of the following components:

3.3.1. Nodes:

The nodes are basically the end stations in the network. Traffic originates from and terminates at the nodes. Nodes basically generate packets that have to be sent to other nodes. Each node is a DEVS atomic model and has the following input and output ports:

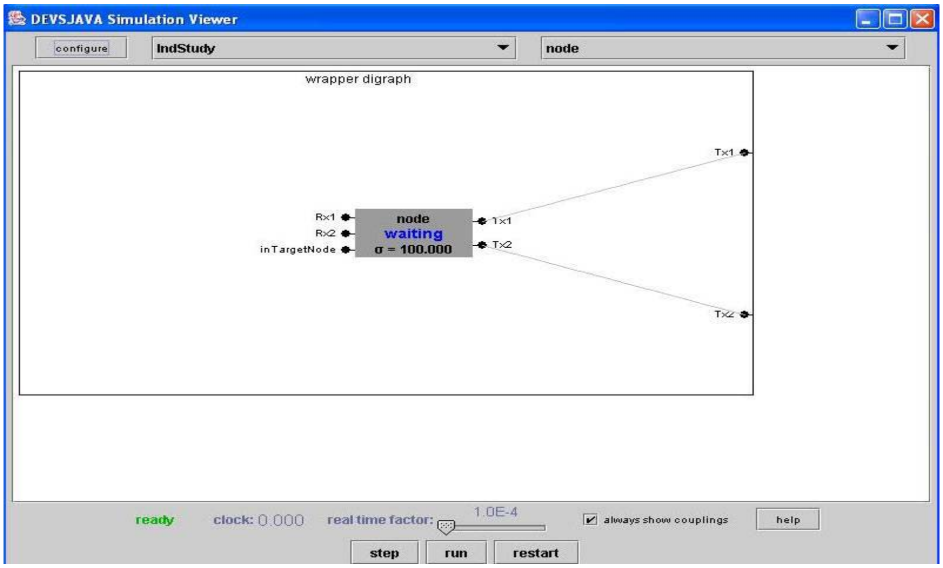

Figure 14. Node

- Input ports (Rx) to receive packets
- Output ports (Tx) to send packets. A node has the same number of input and output ports.
- Target Input port (inTargetNode) to initiate the traffic flow. Packets containing destination node ID, route and size are injected into the node through this port. The node then sends out these packets into the network.

Figure 14 shows the structure of a node. Each node also has a node ID to distinguish it from other nodes.

3.3.2. Links

Links are used to connect together different nodes. A link can be either a point-to-point or a broadcast link. In the former, the link connects only two nodes to each other. A WAN is usually composed of such point to point links.

A link is also an atomic model and has the following input and output ports:

- Input ports (Rx) to receive packets
- Output ports (Tx) to send packets. Same number of input and output ports.
- Capacity which is the maximum number of bytes it can transmit at any instant of time.

Figure 15 shows a link.

The link receives packets through the RX ports and places these into an input Queue. Then it forwards these packets after incorporating delay (or latency). The delay is computed based on the packet size and the capacity of the link. Each link has a link ID to differentiate it from other links.

Each link also has a capacity which is the maximum number of bytes that it can hold at any instant of time. When it holds more bytes than its capacity, it is considered to

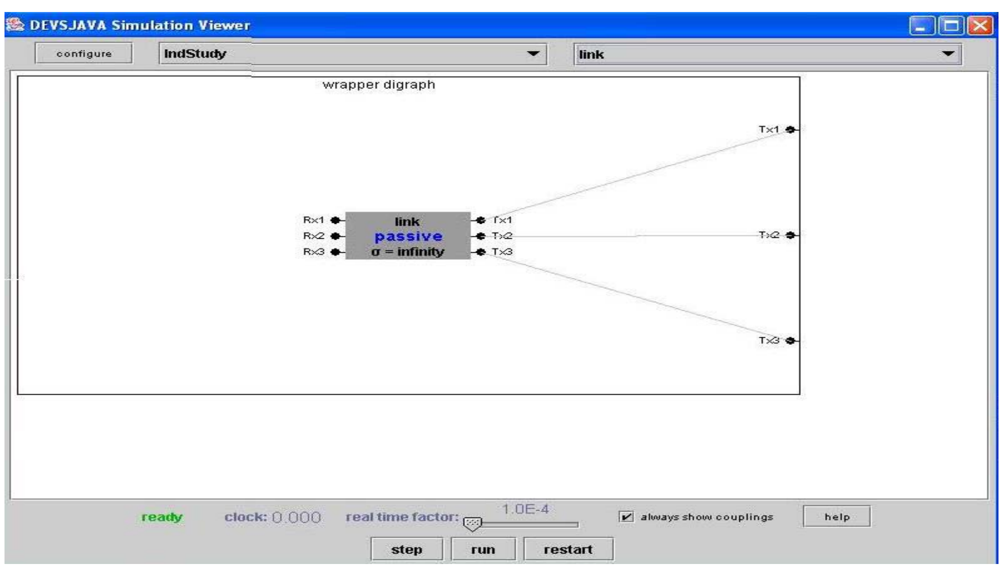

Figure 15. Link

be overloaded. The fraction of the bytes it holds (it is trying to transmit) to its capacity is the utilization factor, u, at that instant. Thus u>1 implies that the link is overloaded.

3.3.3. Subnetwork

A group of nodes and links forms a subnetwork. Different subnetworks may be connected to form networks and higher hierarchical models. A subnetwork is a DEVS coupled model and consists of the following:

- A set of nodes and links connected (or coupled) to each other based on the network topology
- Input ports (In)
- Output ports (Out)

The In ports are used to configure the traffic flow in the network – this means setting up the source nodes, destination nodes, packet size and routes. They can also be used to receive packets from other subnetworks. The Out ports are used to send packets to different subnetworks. Thus, subnetworks can be connected together to form networks and thus develop the network hierarchy.

The procedure to couple nodes and links while forming a subnetwork is explained with an example:

Consider two nodes n1 and n2 connected together using link l1

- Couple Tx1 of n1 with Rx1 of l1.
- Couple Tx2 of l1 with Rx1 of n2.
- Couple Tx1 of n2 with Rx2 of l1.

- Couple Tx1 of l1 with Rx1 of n1.

Figure 16 shows the structure of a subnetwork (an arbitrary subnetwork).

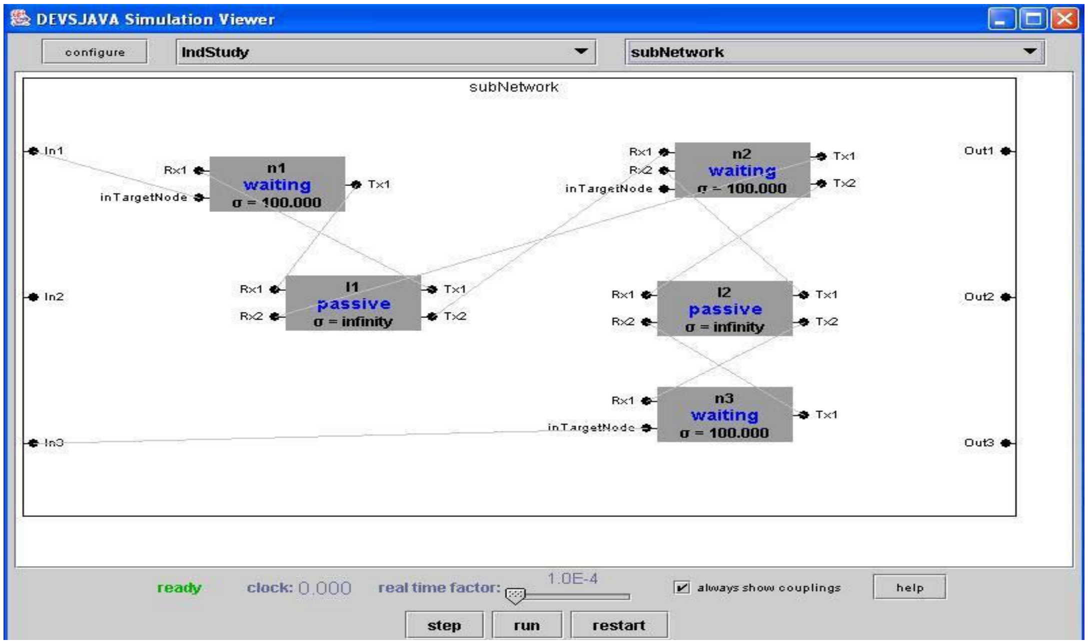

Figure 16. Subnetwork

3.3.4. Packets

Packets are the actual messages being sent from one node to another through a set of links. A packet is not a DEVS model (atomic or coupled) as such. It is a DEVS message. Each packet is a triplet consisting of the following fields:

- A destination node ID
- Route
- Packet size (number of bytes in the packet)

Packet Forwarding:

Packets are forwarded using "Source Routing". This source routing is implemented using the Route field in the packet.

The Route field is a queue that consists of the links and nodes through which the packet has to travel to reach the destination node (specified by its node ID) from the source node. This is actually made up of the output port numbers in each component (node and link) that the packet has to traverse to reach the destination. For example, in the subnet shown in figure 16, consider that a packet has to travel from node n1 to node n2. First, n1 forwards the packet to link l1. To do this, n1 has to send the packet out through output port Tx1. Link l1 has to then send this packet to node n2 through output port Tx2. The packet then reaches node n2 which is the destination. The complete Route queue for this packet is thus [1, 2].

The packet begins its journey at node n1. So the Route field of the packet initially when it is injected into n1 is [1, 2]. n1 checks the first entry in the Route queue and sees 1. It then sends the packet out through Tx1 and removes the first entry, 1 from the queue. When l1 receives the packet, the Route queue is [2]. So, when l1 takes the first element from the queue, it sees 2 and sends the packet out through Tx2 after removing the element from the queue. The Route queue is now [] or in other words, empty. When n2 receives this packet, it sees an empty Route queue and realizes that it is the destination for the packet.

3.3.5. Link Models Developed:

There are three separate models that have been developed:

- In the first model, there is no delay or queuing of packets in the links. In this model, nodes generate packets and the links forward these packets immediately. Thus, the whole simulation takes place in "zero simulation seconds". This model helps in observing the load on each link and detecting whether a link is overloaded or not.
- In the second model, the links store all the incoming packets in a queue. They then forward these packets after a certain amount of delay. If the links receive any packets within this delay, they store the packets in the queue, but the delay remains unaffected. After transmitting the present set of packets, the links then consider the newly arrived packets and transmit them after the appropriate delay. The delay is calculated as 1/(1-u) where u refers to the utilization of the link.
- The third model is a variation of the second model described above. The links store incoming packets in a queue and then transmit them after a delay. If any new packets arrive within this time, the links recalculate the delay and transmit all the packets. The delay is calculated as before.

In each of these models, if at any instant, there are more bytes in a link than its maximum capacity, the link model turns red in color to indicate an overload.

3.3.6. Experiments and Results

For each link, experiments were designed and the link model was tested for two cases: normal operation and overload. In all cases, the delay was correctly introduced into the transmission of packets and when overloaded, the link turned red to signify the overload. Figures 17 to 19 show the three models in their respective overloaded states.

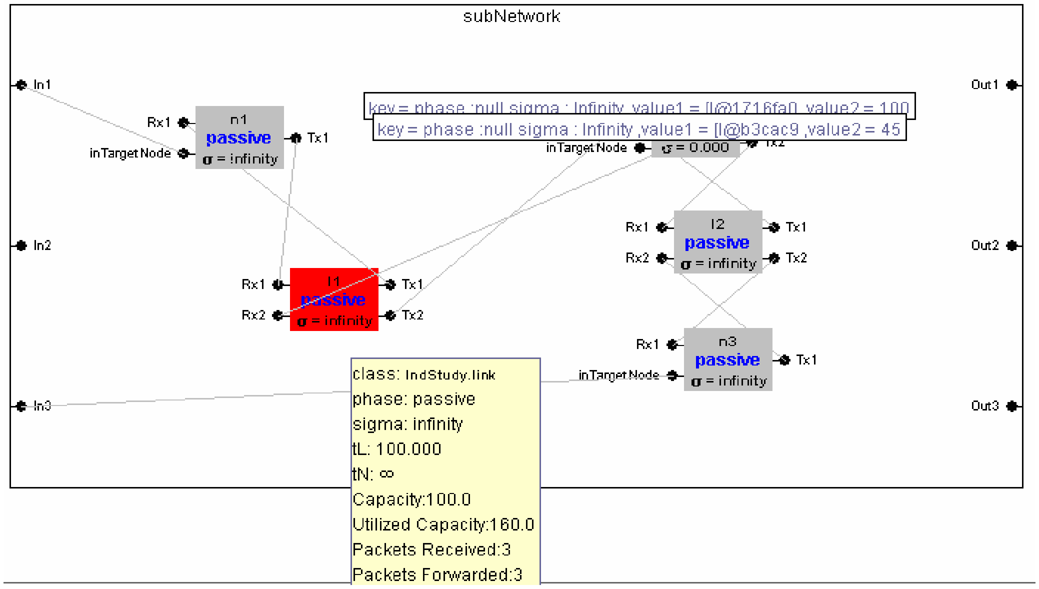

Figure 17. Link Model 1

3.3.7. Advantages of using Abstract Models

All the models developed in this study are abstract models. In real networks, there are a lot of steps involved in processing each packet. However, the models demonstrated above achieved the same end results of transmitting packets from a source to a destination without violating any rules, but at the same time abstracting away all the unnecessary processing details. The three link models described here have different levels of abstraction as compared against a real link in a network.

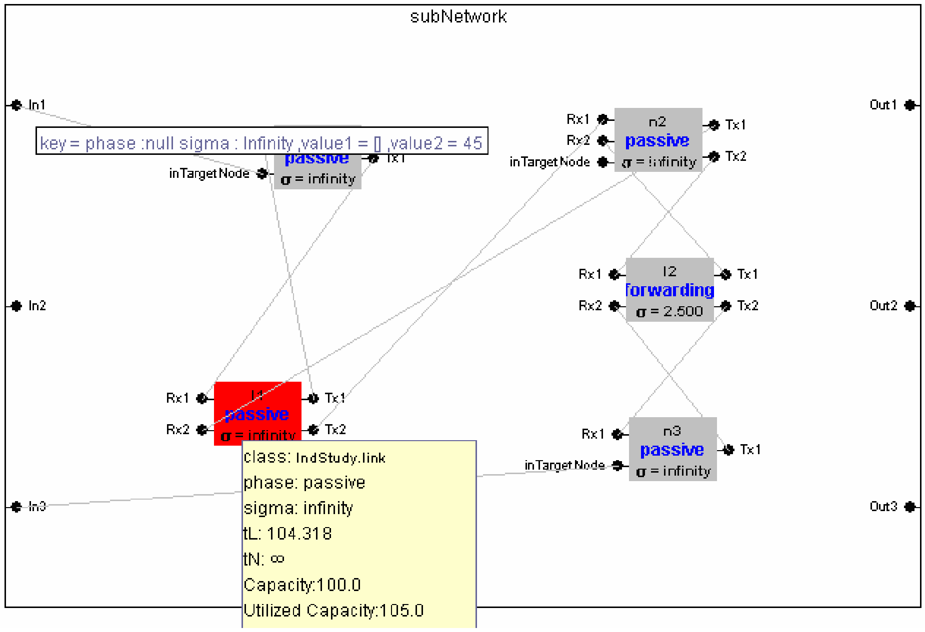

Figure 18. Link Model 2

The first model without any queuing delay does not incorporate many details of the working of an actual link in a network and thus, is a very low level model where as the third model is a more detailed model since it includes some level of queuing delay characteristic of an actual link. The second one is an intermediate detail model since it includes details not modeled in the first one, but does not include as many details as the third one. All three models could be quickly developed and had the same interfaces. So each model could be developed independently of the other models. Abstract models thus have the advantages of being quickly developed, having just enough features necessary to answer the required questions and the flexibility to allow more details to be added.

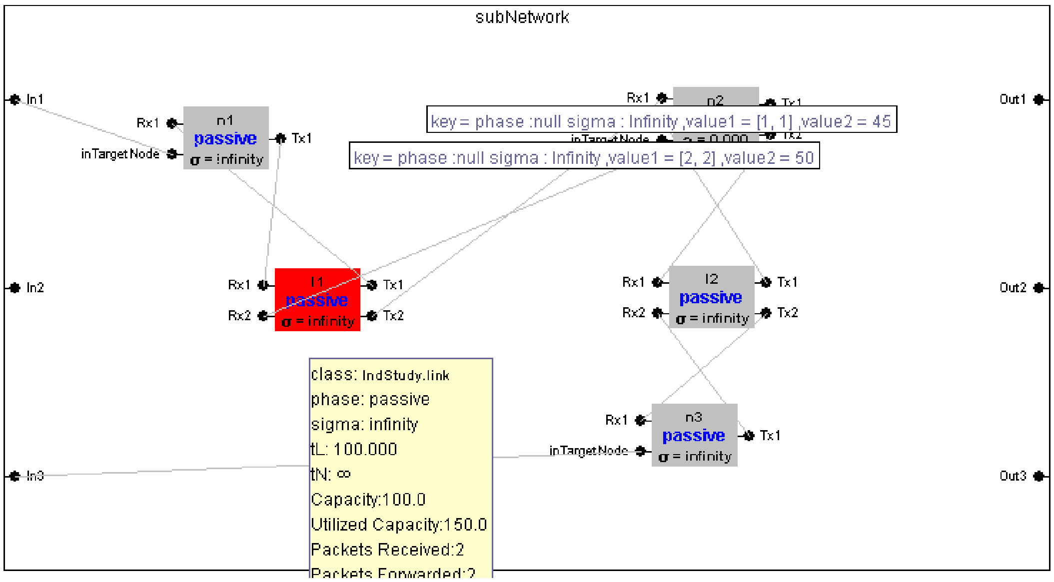

Figure 19. Link Model 3

4. SCOPE Command Simulation Components

In this chapter, the various components involved in the modeling of the SCOPE system are introduced and their behaviors are examined briefly. An architectural framework that was proposed for the modeling of this system is also shown here. The simulation components examined here represent basic behaviors and are used in conducting simple experiments. Based on the results of these experiments, more functionality can be added to these components and additional experiments can be conducted.

4.1. Understanding NetSim

Since the DEVS models are basically an abstraction of the more detailed NetSim models, it is important to gain an understanding of NetSim. Only a very brief explanation is given here.

4.1.1. Structure

Understanding the different components of NetSim and their structural relationships is greatly facilitated by the System Entity Structure (SES) diagram shown in figure 20. At the topmost level, the various components in the system are: Traffic, Entities, Stations, Channels, Events and the StateRec (stands for State Record).

The Traffic component or entity has various parameters or attributes such as rate of transmission, size of the transmission structure, source station, destination station and so on. The Entities component is made up of several individual Entity components (the name of this component itself is Entity. It should not be confused with the term “entity” used in the SES). Each Entity has two attributes: a station and a StateRec. An Entity is a general structure and hence, has various specializations such as Station Entity, Traffic Entity and so on.

The Stations component is made up of several Station entitis. Each Station entity has attributes such as Traffic and Activity Type. Figure 21 shows a separate SES of the Station entity. It can specialize as a Mobile and Fixed Station. A Fixed Station can also be refered to as a Ground Station and these two terms are used interchangeably in this report.

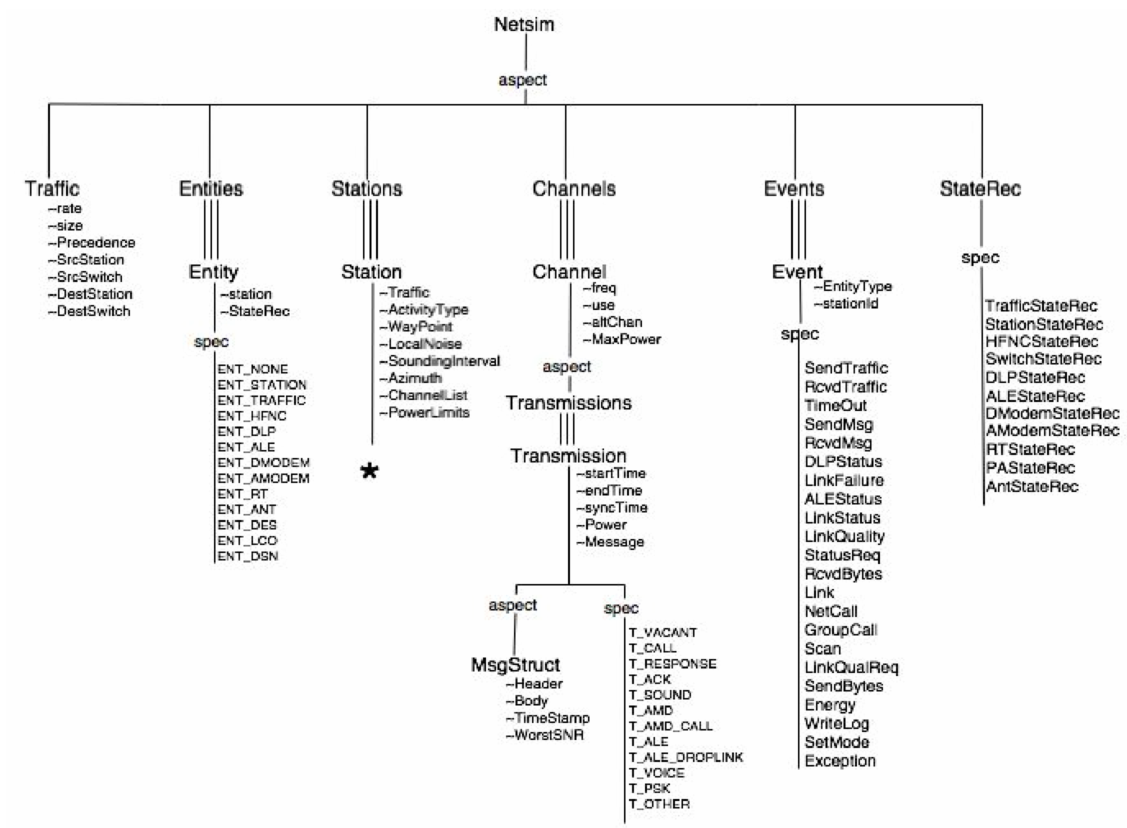

Figure 20. NetSim System Entity Structure

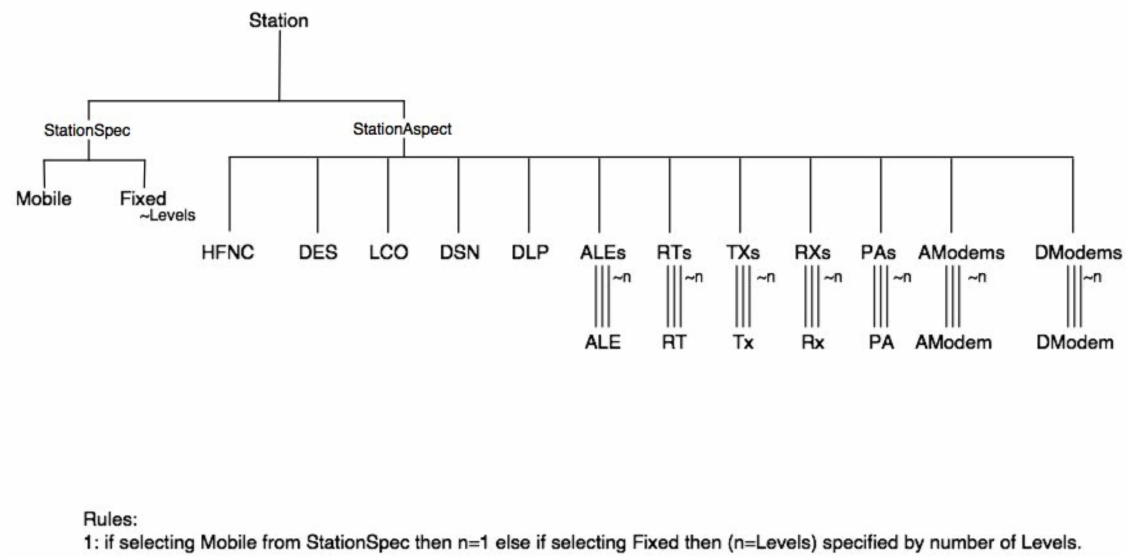

Figure 21. NetSim Station SES

Each Station also comprises a HFNC (HF node controller), DES (a switch), LCO, DSN, ALEs, RTs, TXs, RXs, Pas, AMode. ALEs is in turn made up of many ALE entities. RT refers to a Receiver Transmitter pair, TX is a transmitter, RX is a receiver, PA is a power antenna and AMode is the antenna mode. In the SES, n refers to the number of levels, that is, the number of ALE radios in each station. If the station is a mobile station, then n = 1, else n is decided by the system design. Each ALE radio is considered as an ALE level, and so in future it will be referred to as an ALE level.

The Channels entity is made up of many Channel entities. Each Channel has attributes such as frequency, use, alternate channel, maximum power, etc. Each channel also contains a Transmissions entity which in turn is made of Transmission entities. A Transmission has attributes such as start time, end time, power, etc. It contains a MsgStruct entity and has various specializations such as call, response and others. A MSgStruct has header, body, time stamp and worst SNR attributes.

The Event entity is a very important part of the NetSim structure. It has two attributes, the entity type (this entity refers to the Entity component of NetSim) and the station id. It has various specializations as can be seen from the figure. Basically, any action in NetSim, whether within a component or across components, takes place through an Event. The initiator of the action sends an Event and the actual component which has to act, acts upon this Event and produces the next Event which turns on some other component and the chain continues.

The StateRec entity serves basically to record the state of any entity in the system. Accordingly, it is specialized into various versions for almost all entities discussed above.

Both kinds of stations – mobile and fixed have the same structure with the difference that a ground station can have more than one ALE level and other associated components, while each mobile station has only one level.

4.1.2. Working

The version of NetSim received for this project at the ACIMS laboratory consisted of a detailed implementation of the ALE protocol. First, all the ground stations and mobile stations are entered into a configuration file. This would specify the number of ALE levels at each station and the flight path for each mobile station (aircraft). The mobile stations would also be given the set of ground stations to connect to (in some order of preference) whenever there is a need to do so.

The mobile ALEs send sound signals at every sounding interval. When the ground station ALEs receive this sound transmission they get the LQA and update their LQA databases. LQA is based on the SNR, so in essence it is the SNR that is to be calculated. NetSim uses the ICEPAC data in order to get an accurate value of the SNR of that particular signal. When a mobile station decides to make a call, its ALE sends out the scanning call and the three way handshake is followed. The SNR values are fetched from the ICEPAC and the LQAs are calculated for the calling signal and the calls are accepted based on this LQA score. In calculating this SNR, NetSim takes into account transmitter power, local noise at the stations, transmission power, antenna mode and other factors. Calculations are fairly comprehensive and adhere very closely to the real world situations. Further, when messages have to be sent,.they follow the whole procedure of going through the modem, DES switch, RT and the antenna.

When there are several messages on the same channel (irrespective of the source and destination), only the strongest signal is chosen as the signal and all others are treated as noise. Strength is determined in terms of SNR derived from ICEPAC and adjustments made to this value based on the other factors mentioned above. The effective strength of the strongest signal is recalculated in the presence of all the noise and delivered to the destination. So, effectively, on a given channel only one transmission takes place in the entire system.

4.1.3. Need for Migration

Before examining the DEVS models, it is important to consider the reasons for a migration from NetSim to DEVS. Some of the factors motivating such a migration are:

- Technology used to build NetSim is outdated
- It is hard to maintain and upgrade the NetSim model

- Several system upgrades are planned over the next few years.

Therefore a new model has to be developed which can answer the questions of the SCOPE System designers.

The NetSim model placed emphasis on modeling the correct working of the ALE. It modeled the working of the ALE protocol in detail. The model also incorporated the actual flow of data through all hardware devices such as the modem, antenna, etc. The main of this model was to check whether a link could be established from an aircraft to any ground station while the aircraft was in motion.

This model in its present state it could not answer the questions about system utilization, the effects of adding/removing resources in the system and other questions that the designers had when faced with the task of upgrading the system. Thus the DEVS models developed as part of this study focused on providing quick answers to such questions.

The DEVS models built here are abstract models in that they model only the essential details of the ALE protocol. They also do not place emphasis on actual power levels and other hardware details. The intent here is to build models with the ability to incorporate more details in them when necessary. The models built here provide answers at a higher level of abstraction of system details. If necessary, more details can be added to these models easily.

4.2. DEVS Architecture for the Simulation

The proposed architecture of the SCOPE System deals with the mobility of the planes (mobile radios) and the ALE Controllers at all radios (mobile as well as fixed). In this architecture, the fact that the only reason the ALE controllers need to know the position of the mobile radio, time of day, month and year is to access the ICEPAC data to calculate the SNR is exploited. The fetching of SNR values from the ICEPAC data base is now made a part of the Experimental Frame. The EF supplies the ALE controllers with the SNR values whenever required.

With this scheme, the mobility of the plane can be reduced to just having a file with:

- required SNR/LQA scores at different times or
- position of the plane (co ordinates) at different times

The proposed architecture is shown in figure 22.

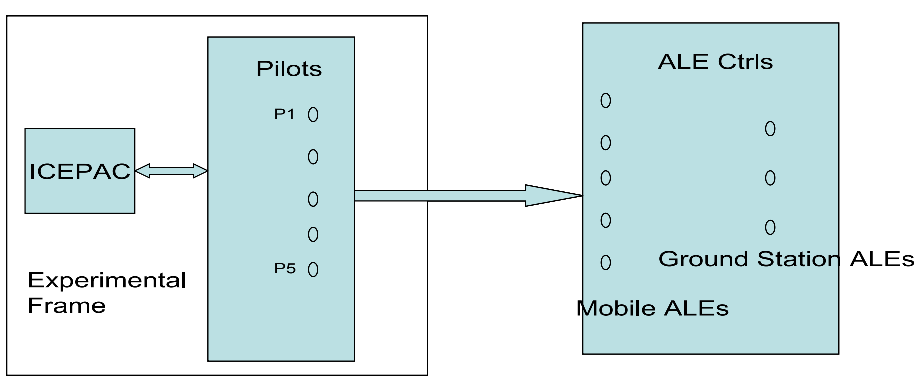

Figure 22. Proposed Architecture for modeling the SCOPE System

The architecture essentially divides the system into an Experimental Frame (EF) and the ALE Controller system. The Experimental Frame (EF) consists of the pilots and the ICEPAC data. The pilots basically represent the mobility of the planes or the position of the planes at any instant. The ALE Controller system consists of the ALE Controllers in the planes as well as those in the ground stations.

Thus an aircraft is divided into two parts:

- mobility – represented by the pilot
- communication – represented by the on board ALE Controller

4.2.1. Experimental Frame

In this section, the experimental frame is explained in greater detail. Figure 23 shows an expanded view of the EF.

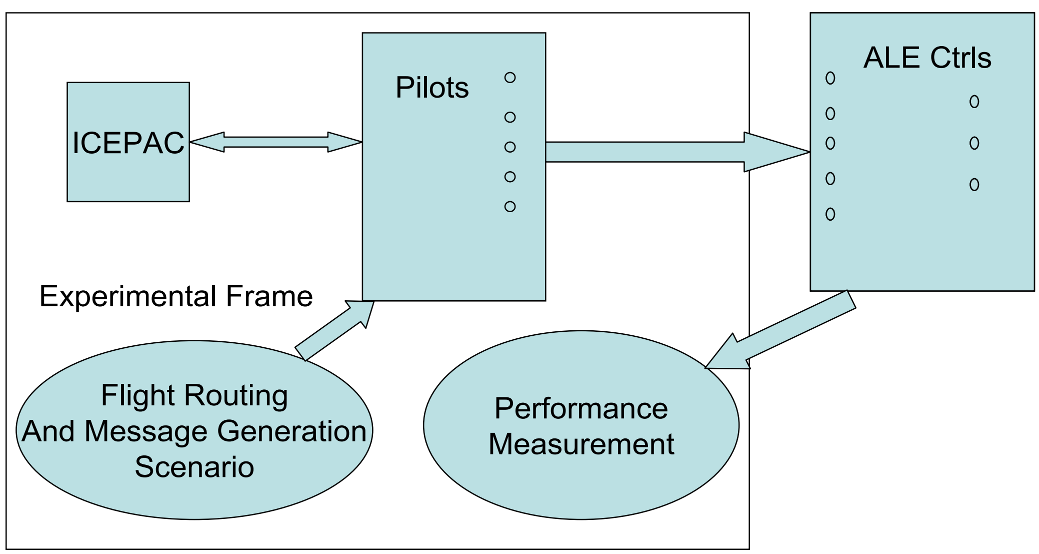

Figure 23. The Experimental Frame in more detail

The EF consists of two components – the generator and transducer. The generator is the component that generates different experiments and drives the system. The transducer on the other hand, receives different kinds of measurements from the components. The transducer can be examined to find out how each component performs or the capacity of each component and other similar statistics. In this architecture, the Flight Routing and Message Generation Scenario component functions as the generator and the Performance Measurement component functions as the transducer.

4.2.1.1. The Flight Routing and Message Generation Scenario component has two important functions:

- Generate the flight plans, for each aircraft in the system.
- Generate messages from the aircraft to the ground stations

Generating the flight path may be based on different factors. Some of these factors are shortest path from source to destination waypoints, path having the best connectivity to the ground stations, etc. For example, consider an aircraft that starts from Location A and needs to reach Location B. The shortest path for such a flight is, say, through points C, D and E. Along this path, suppose there are two Ground Stations 1 and 2. So, one possible flight plan is to have the plane start from A, go through C, D, E and reach B. Along the way, if the plane needs to communicate, it will communicate or try to communicate to GS 1 and 2. Now, consider a scenario in which GS 1 and 2 are usually busy during certain periods. So if this plane tries to communicate with these stations, it might not be possible to establish a link. Suppose there is an alternate path which involves flying through locations, F, G, H and I. Along this path, there are stations 3, 4 and 5. Although this path is not the shortest path between A and B, it has the advantage that the probability of establishing connections with Stations 3, 4 and 5 is higher than establishing one with Stations 1 and 2. So, a new flight plan would be to have the plane start from A and go through F, G, H and I and finally reach B. Along the way, if it needs to send/receive any messages, it communicates through Stations 3,4 and 5.

The message generation part would involve a conventional random generator which generates messages with an inter arrival time following usually an exponential distribution. Uniform random generators may also be used, although frequently in performance measurement studies, the exponential generator is used [14]. In systems involving queuing, congestion, etc, it is found that the number of jobs arriving or completing service in a fixed interval of time is approximately Poisson distributed. Therefore the inter arrival time of these jobs is exponentially distributed. Hence, an exponential generator is used instead of a uniform generator.

<u>4.2.1.2. The Performance Measurement</u> component receives various measurements from the different components in the system. For example, it can receive measurements, such as number of calls made, number of calls dropped, number of packets/bytes sent/received from different ALE Controllers, the Ground Stations and the CNCS station. Using such measurements, various performance measurement factors may be measured such as:

- Total number of packets/bytes sent and received by each ALE Controller, ground station and the CNCS station
- Total number of calls/messages received/made at each Ground Station
- Total number of calls dropped at each Ground Station.

These factors can then be used in making different flight plans and thus generate different kinds of experiments in the Flight Routing and Message Generation Scenario.

This set-up provides the facility to have different message generation scenarios in the experimental frames:

- Send only/Receive only: In the send only scenarios, the planes only send information to the ground stations. In the receive only scenarios, only the ground stations send messages to the planes.
- Send and Receive: In these scenarios, the planes as well as ground stations can send messages to one another.

Thus, the EF can be used in generating various experiments and collecting different kinds of performance measurements.

4.2.2. Modeling of Voice and E-Mail messages:

The effect of any communication, whether voice or e-mail is dealt as using a channel and making it unavailable. Thus, both voice and e-mail messages are treated as the same. In practice, the E-Mail messages go through the STANAG 5066 protocol stack and then are exchanged through the ALE links, while voice calls are directly sent over the ALE links. This implies a time delay in the processing of HF E-mail messages. Additionally, ALE calls meant for exchange of HF E-mails are linked longer than typical voice calls. Use of STANAG 5066 also implies additional traffic in the system due to the exchange of acknowledgements. However, this could be modeled by setting appropriate message sizes. Since the E-mail calls tie up links for longer times, fewer channels would be available at any given time for other e-mail or voice calls. Typically, certain channels are set apart for voice calls and certain others for e-mail calls.

Since the focus in this research is on building abstract models and observing the overall performance of the system, a distinction is not made between voice and e-mail calls. All ALE calls are treated in the same manner. Of course to model e-mail calls, the aircraft and generator models could send more messages as in typical e-mail scenarios, but they do not do so in the experiments in this research. Although models of the STANAG 5066 protocol were developed, they would be better placed in a detailed model which would also contain the exact modeling of the ALE protocol with all phases and message exchange details and so on.

Also since the purpose here is to look at the architecture and abstract models more as a development methodology and approach to planned expansion rather than actual experimental indications for any expansion of the SCOPE system, the distinction between voice and e-mail calls was not considered significant.

4.2.3. Advantages of this architecture:

An important use of this architecture is that the mobility part of an aircraft is separate from its communication part. The ALE systems of both the mobile stations and ground stations can be grouped together into the ALE Controller System. It is easier to generate different experiments dealing with the mobility of the aircraft (that is, different flight paths) and different message generation scenarios. Since the ICEPAC data is now part of the EF, the EF can also find the correct SNR value and update the ALE controller. In the same way, the development of ALE models is now independent of how the plane's mobility is taken care of or how the ICEPAC data is accessed. This allows the development of different sets of ALE models with the required level of detail or abstractions. Thus, for running experiments which basically need to measure the performance of the system, more abstract models of ALE can be built quickly while for more detailed observations of the system and to find out how links get established, more detailed models of ALE can be developed and used.

Further, another advantage of architecture is that the EF instructs the ALE which station to connect to and which channel to use. This in turn leads to the fact that the station and channel selection algorithm is part of the EF. These different algorithms may then be used in various experiments and some guidelines on how the system must be used can be found. Additionally, some other questions of the SCOPE system designers can also be answered. For example, the designers can now prepare a list of ground stations that a mobile station will attempt to connect to. These stations will be arranged in order of preference. The designers can also find out the effects of adding a new station or a new channel and on the same note, removal of an existing station or channel or any combination of these.

<u>Best SNR Selection:</u> In this algorithm, the aircraft, based on its current location and other factors gets the SNR from ICEPAC for all the ground stations on each channel in the system. It then chooses the station, channel combination yielding the best SNR. In case this does not work, the combination yielding the next best SNR is tried and the process continues.

Best Station, Channel Combination: In this algorithm, each station is first considered. The channel that yields the best SNR for this station is chosen as the one to use. This yields different station, channel pairs in which the station is unique. These pairs are then ranked in order of the SNR. This is different from the previous algorithm in that, this may not be the actual order of best SNR since each station is allowed to appear only once. This algorithm can then be used to answer questions about the addition of new ground stations and the like.

Best Channel, Station Combination: This algorithm is similar to the previous one except that the roles of the channel and station are reversed. Experiments based on this algorithm can be used to answer questions regarding the addition of new channels.

4.3. System Components

Having examined the architecture used for the modeling of the system, it is time to describe the actual components that are used to implement this architecture.

The basic system components of the SCOPE Command system that need to be modeled are:

- Aircraft
- ALE Radio

The modeling of an aircraft implies modeling its movement. Every aircraft in the system is given a flight route, which is basically the set of airports that the aircraft passes through. In the rest of this document, these airports are referred to as waypoints. An aircraft always moves between two waypoints over the great circle. Due to the shape of the Earth, when planes or ships move between two points, the shortest path between the two points is given by the great circle path and not necessarily the straight line path between these two points. Whenever required, the position of the aircraft can be calculated based on its trajectory along the great circle.

The modeling of the ALE radio implies modeling of the ALE protocol. As shown above, NetSim builds a very detailed model. However, the DEVS models developed here model these ALEs at an abstract level. Whenever a connection has to be established, the transmitter and receiver ALEs follow the three way handshake protocol. Since the SNR values are fetched from the ICEPAC and sent to the ALE controllers, there is no need to implement the sounding and building a database. The sounding process does not interfere with the calling or connecting processes. Since the EF uses SNR values from the ICEPAC and use them for station, channel selection purposes, there is no need of an LQA database. If the LQA database is not needed, then there is no need to model the sounding process. Even the timing details for the link establishment process are modeled at an abstract level since the main point of interest in these experiments is not latency.

In addition to these basic components, certain other components need to be designed. They are discussed in the following sections.

4.3.1. Waypoints

Each waypoint is built as an atomic model. A waypoint basically has a position coordinate and a time associated with it. The position coordinate is made up of the latitude and longitude at which the waypoint is located. Since a flight route is made up of one or more waypoints, each waypoint other than the first one or the starting point is offset from the previous one in time. This time offset is the time taken to travel from the previous waypoint to the current waypoint. The time attribute of this waypoint is this offset. For example, if a flight route has the aircraft starting at WP_1, go on to WP_2 after traveling for time t_{12} and then reach WP_3 after traveling for time t_{23}, the time attribute at each of these waypoints would be:

WP_1 = value configured by system designer

$WP_2 = t_{12}$

$WP_3 = t_{23}$

WP1 has the offset as a value configured by the system designer since that would constitute the time at which the aircraft begins its journey. This is usually set as the time at which the simulation starts.

The set of waypoints in the system are entered into a configuration file or a waypoint file. The EF has to first read this file and then store all the waypoints that exist in the system.

Each waypoint is considered as a router in a network. Consider now the set of waypoints as a set of routers in a computer network. Similar to routers being connected to one another, waypoints can be connected to one another. This connection of waypoints to one another may be based on various criteria:

- Distance: The most common rule for connecting waypoints to one another would be the distance between them. Two directly connected waypoints would imply that an aircraft could fly between these points in a single flight segment. So//,// all waypoints say, within distance d of each other would be directly connected. To travel between any two waypoints not directly connected to each other would involve finding a path through a set of connected waypoints between these two.

- Propagation conditions: This would be a situation wherein adverse weather conditions would render a flight along that path impossible. Therefore, only waypoints which experience good flight conditions would be connected to each other.

These are but two of the several criteria for connecting waypoints. However, all these may be considered as a modification of the first one, that is, the one based on distance. The calculation of the great circle distance is not explained here. [15] – http://www.codeguru.com/Cpp/Cpp/algorithms/general/article.php/c5115/ - provides a good reference for many calculations related to navigation.

The advantage of considering waypoints as routers and connecting them is that, a flight route now can be generated using various Network Routing Algorithms. For example, a flight route that needed to be the shortest path can be generated by running the Dijkstra Algorithm [16] on this network to build routing tables at each waypoint and finding the shortest path using these tables. Dynamic routing can be achieved by breaking links between waypoints based on bad weather conditions and so on (similar to links going down in computer networks).

Figure 24 shows an individual waypoint and figure 25 shows a network of connected waypoints.

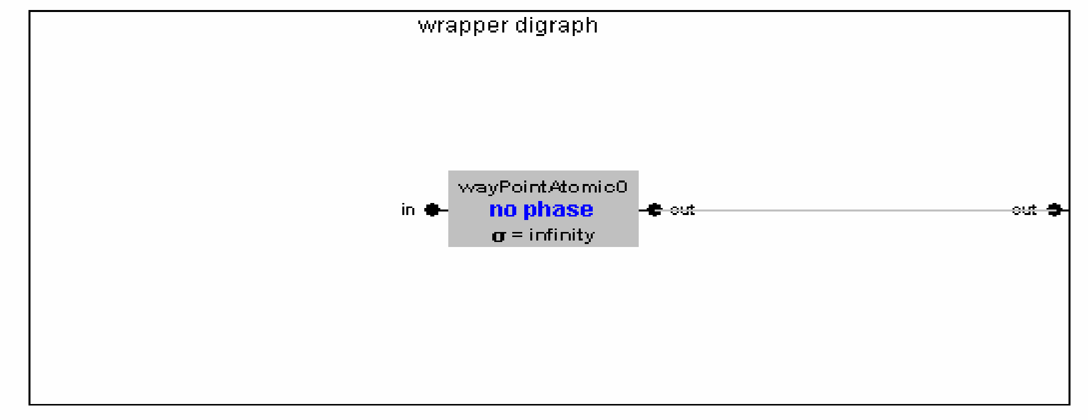

Figure 24. Individual Waypoint

In this configuration for example, the flight route between WP0 to WP2 is WP0 -> WP1 -> WP4 -> WP5 -> WP2. In the experiments conducted as part of this research, the waypoints are connected based on the great circle distance between the waypoints.

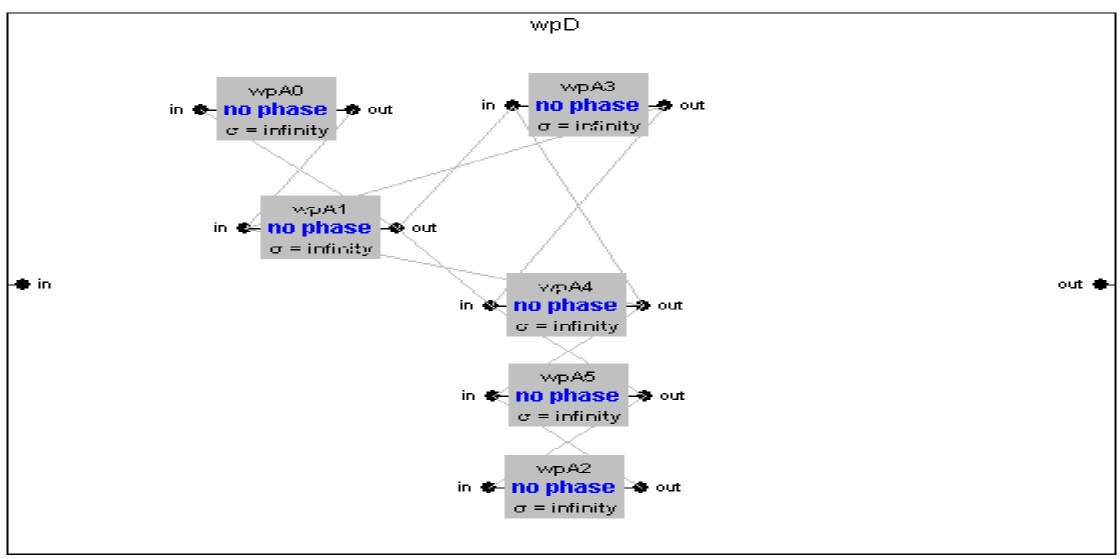

Figure 25. A Waypoint Network

4.3.2. Aircraft

There are two main functions involved in the aircraft: mobility and message transmission.

The movement of the aircraft depends on the flight route that is specified. Each aircraft starts at the initial position or starting point referred to as WP_0 (WP stands for waypoint). Over time, it reaches the next waypoint WP_1, from there goes to WP_2 and so on until ultimately it reaches the destination. In a DEVS model, this transition is treated as an internal transition function. The following algorithm explains the whole procedure:

1. Start at the initial position. Stay in the initial position phase until it is time to be at the next waypoint. At this point enter the next waypoint phase.
2. While (destination not reached)
3. Stay in current waypoint phase until it is time to reach the next waypoint. Then enter the next waypoint phase.
4. If this is the destination go to 5. Else go to 3.
5. Stay at the destination until the end of the simulation.

Figures 26 - 29 show an aircraft in different phases in its journey.

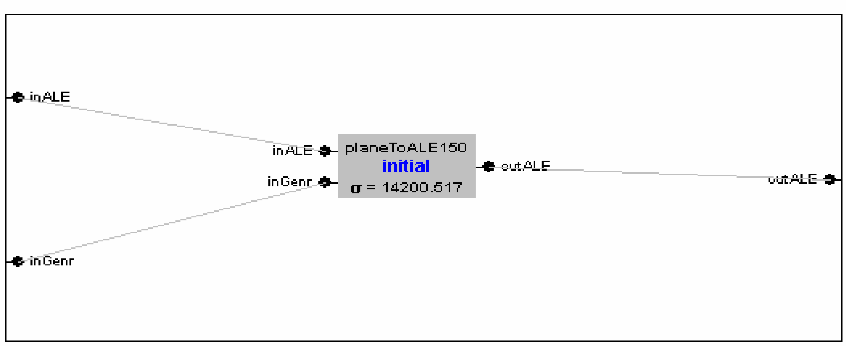

Figure 26. Plane in initial position or between initial and WP1

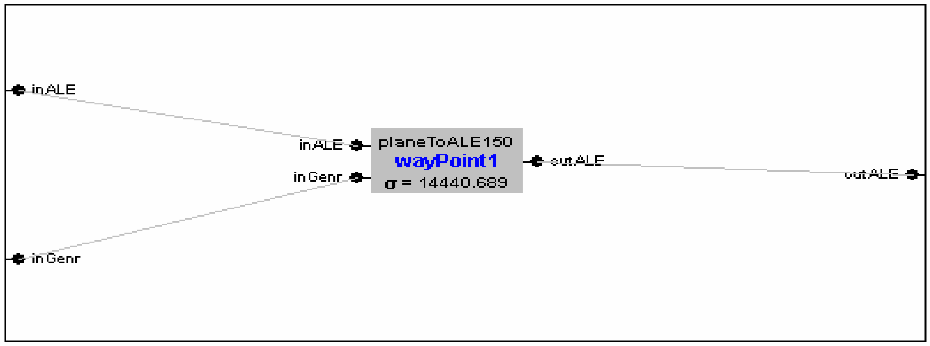

Figure 27. Plane at WP1 or between WP1 and WP2

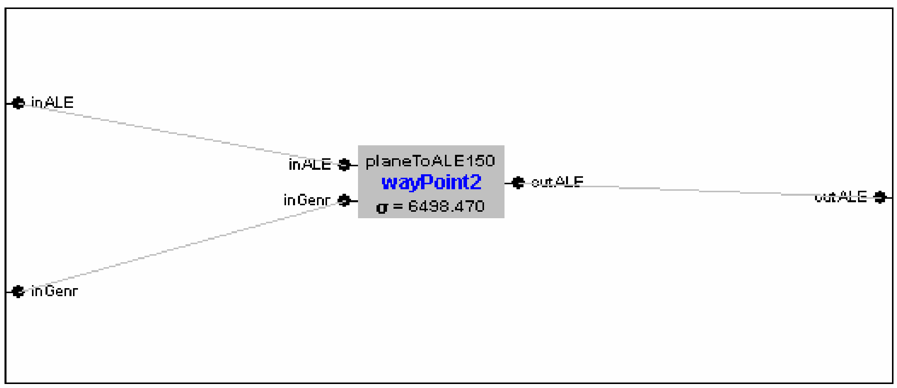

Figure 28. Plane at WP2 or between WP2 and Destination

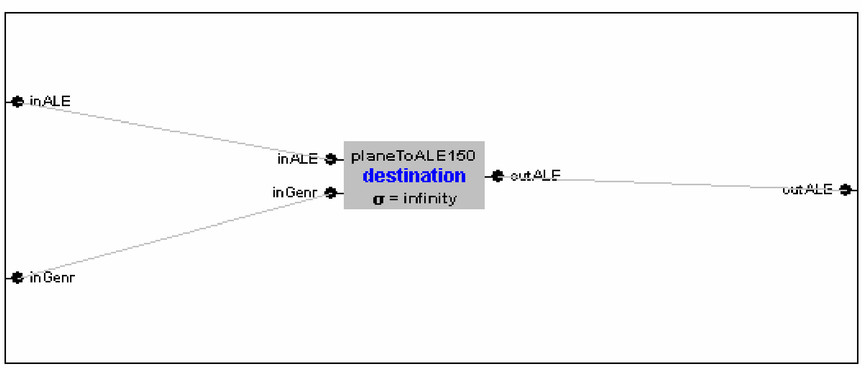

Figure 29. Plane at Destination

The aircraft can be triggered to send a message any point in its journey. The aircraft at this point calculates its present position and tries to find the best station to transmit to and the corresponding best channel to use for that station. This is done by using the SNR values accessed from the ICEPAC database. It then sends a message to its corresponding ALE to connect to that station on that channel and also gives it the SNR value which is to be passed on to the receiver ALE for evaluation purposes. It then continues its flying until it receives a message from its ALE about the link establishment. If the link has been established, the aircraft sends its ALE the message to be sent to the receiver. Possible courses of action if the ALE reports that a link could not be established include:

- The simplest action would be treat this message transmission attempt as a failure and continue flying.
- Alternately, it could find the next best station, channel combination to transmit. This choice could be made based on any of the selection algorithms highlighted in section 4.2.2.

The calculation of the exact position of the aircraft is a simple procedure if the flight path is known. The interested reader can find the algorithm in any good reference on navigation such as [15].

In the experiments conducted in this research, the aircraft tries once to transmit its message. If that attempt is unsuccessful, it gives up.

After examining the functional behavior of the aircraft, it is now time to take a look at its structure. The aircraft is constructed as an atomic model. The DEVS aircraft model is shown in figure 30.

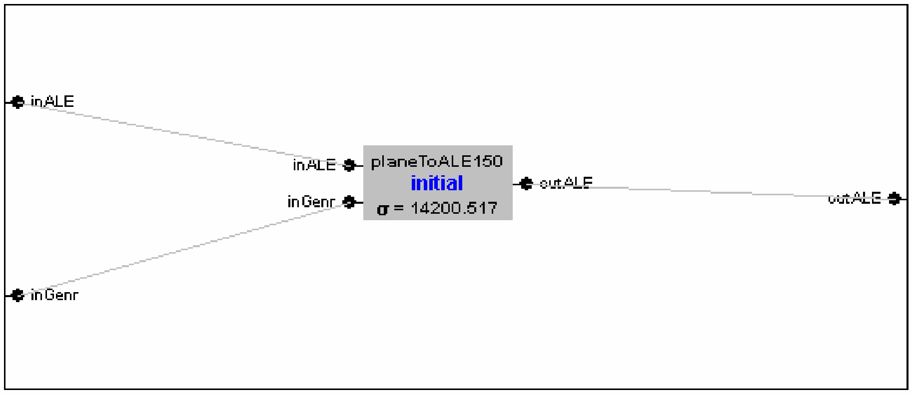

Figure 30. DEVS Aircraft Model

Each aircraft has two input ports “inGenr” and “inALE” and one output port “outALE”. These are the ports through which the aircraft interacts with the other models. It receives the inputs from the generator through the “inGenr” port. When it receives this input it goes into the messages transmission phase which includes calculating the SNR and consequent interaction with the ALE. It sends out messages to the ALE through the “outALE” port and receives messages from the ALE through the “inALE” port.

4.3.3. Generator

The generator used in the experiments for this research is an exponential random generator. Each aircraft in the system is associated with a generator. The generator triggers the aircraft to send a message to a ground station. Thus with the generators triggering the aircraft at time intervals following an exponential distribution, the number of messages generated by the aircraft follows a Poisson distribution. In most computer system performance tests, the Poisson message arrival probability is used as a good system modeling practice.

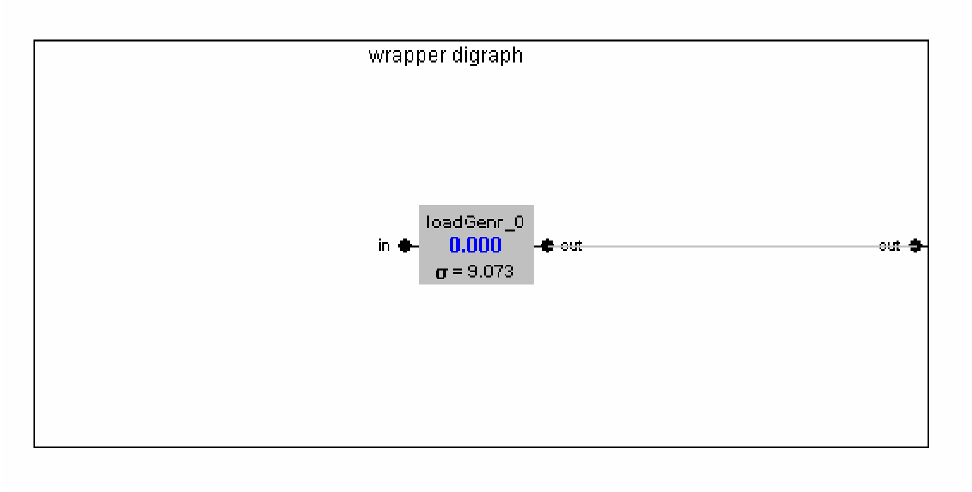

Figure 31. Generator

The generators are provided with a mean inter arrival time. This mean time is configured by the experimenter. Different sets of experiments would use different mean values, for example, 10 minutes or 30 minutes and so on.

4.3.4. Flight Routing

The flight routing is not exactly a DEVS component by itself, but is an important concept in the system. Since different flights traveling over essentially the same paths would lead to the selection the same station, channel combinations, it is important to choose a suitable flight routing scheme.

Entering the flight route information for each aircraft depends on how the aircrafts themselves are put into the system. The number of aircraft can be specified in two ways. The simplest scheme would be to enter all the different aircraft in the system in a configuration file. At the same time, their flight routes could also be input. This is the scheme that is used in NetSim. This method is certainly appealing when the number of aircraft in the system is reasonable (maybe up to 10 or 30). It would be easy to set up a good mixture of conflicting as well as non intersecting paths for the different aircraft.

However, when the number of aircraft in the system is huge, it is not a feasible or practical method to enter all the information in a configuration file manually. In such a case, it is better to generate the flight routes randomly. The user would enter the number of aircraft in the system and the assignment of flight routes is done randomly. This is similar to the Random Waypoint Mobility Model [17] that is frequently used in the routing protocols for mobile networks especially MANETs.

In the experiments conducted as part of this research, only the source and destination are chosen randomly. Once these are known, the route between these two points is decided by the Dijkstra Algorithm or in other words, the shortest path between them. Since all the waypoints are known (see section 4.3.1), a uniform random selector can select the source and destination from this waypoint list. Additional restrictions could be placed such as that every source and destination chosen should have at least one hop between them (that is they are not directly connected). It is also important to note another possibility. Based on the maximum distance between two waypoints that can be connected, there may be certain waypoints that are not connected to any other waypoint. Also the whole network could actually be a set of disjointed networks. So, when a source and a destination are chosen, it should be ensured that there is a path between them.

The generator and flight routing components together form the Flight Routing and Message Generation Scenario component of the EF.

4.3.5. Transducer System

The transducer is the part of the EF which collects the performance measurements. Since most of the statistics need to be derived from the individual ALE radios which are part of the ALE model (or system model), it is better to place a transducer at each ALE. But, ultimately, it is the responsibility of the EF to aggregate the measurements at each transducer. Therefore, there needs to be a transducer in the EF which can do this job. Thus there are two types of transducers in the system: the ALE Transducer and the System Transducer. These are explained in more detail in sections 4.3.9 and 4.3.11.

4.3.6. ALE Components

The ALE radios are made up of many individual components. In DEVS terminology, an ALE radio is a coupled model consisting of atomic models such as ALE coordinator, ALE levels and ALE transducers. A single ALE (or ALE controller) essentially consists of one coordinator, a number of ALE levels and one ALE Transducer. The ALE levels refer to the ALE radios that actually establish connections and transmit and receive messages. The ALE coordinator decides which levels are to be used for a given connection and thus activates the individual levels. This can be seen as the function of the controller in the real ALE system. The ALE transducer is responsible for gathering performance related measures in that ALE for all levels.

4.3.7. ALE Coordinator

The DEVS model for an ALE coordinator is shown in figure 32.

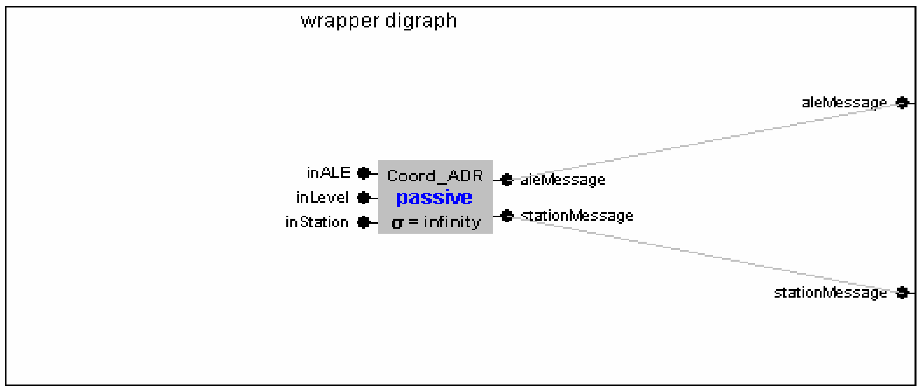

Figure 32. ALE Coordinator

It consists of three input ports, “inStation”, “inALE” and “inLevel” and two output ports “stationMessage” and “aleMessage”. The coordinator receives messages from its station through the inStation port. It sends out these station messages to the appropriate level through the stationMessage output port. It receives messages from the other ALEs in the system through its inALE port and sends these messages to the required ALE level through the aleMessage port. When an individual ALE level sends update messages to the coordinator, such messages are received through the inLevel port.

The behavior of the coordinator is examined under two different conditions: when it is a transmitter and when it is a receiver. It should be noted that, at any instant, it can be functioning as both on different connections provided that the ALE has enough levels to establish the required links.

4.3.7.1. Transmitter:

The coordinator functions as a transmitter when it receives a message from its station to establish a connection with a desired destination on a particular channel. Since only one connection can be established on a given channel between any two ALEs, the coordinator first checks if the channel is available for use. This channel availability is checked at its local database. If the channel is available, it then checks whether an ALE level is available for link establishment. ALE levels are numbered as level 1, level 2 and so on. It starts checking from the lowest numbered level. That is, level 1 is given precedence over level 2 and so on. If an ALE level is available, it passes on the connection request to that level through its stationMessage port. If no level is available, it goes back to the passive state. If the channel itself is not available, then the coordinator goes back to the passive state.

If the station sends any other message for example, a message to terminate the connection, the coordinator passes on the message to the ALE level without any processing.

The algorithm for the working of the coordinator as a transmitter is provided below.

1. *Start in the passive state and stay for infinity.*
2. *if (receive message from station) go to step 3*
3. *if (station id == coordinator id)go to step 4. else go to step 9*
4. *if (message is connection request) go to step 5. else go to step 8.*
5. *check if channel to be used is available. If it is go to step 6. else go to step9*
6. *check if ale level is available to establish required link. If it is go to step 7. else go to step 9.*
7. *update Level Map with Ale level and channel. Also update channel availability*
8. *send station request to the ale level through the stationMessage port.*
9. *go back to passive state.*

4.3.7.2. Receiver:

The coordinator functions as a receiver when it receives a connection request message from another ALE through its inALE port. When the coordinator receives a message through its inALE port, it first checks the destination field in the message and if there is a match with its own address, it processes the message further. Otherwise, it discards the message. If that message is a connection request, it checks the SNR value in that message and if that is above the threshold (for accepting connections), it retains the message for further check. It then checks the channel availability. If the channel that the message uses is already in use, then it discards the message. Otherwise, it checks if an ALE level is available for accepting the connection and if it finds one, sends it the message through its aleMessage port. If the SNR is below the threshold, the message is discarded. If the message was not a connection request in the first place, it is simply sent out through its aleMessage port to the ALE levels.

The algorithm for the working as a receiver is given below:

1. Start in the passive state and stay for infinity.
2. if (receive message from another ALE) go to step 3
3. if (destination == coordinator id)go to step 4. else go to step 10
4. if (message is connection request) go to step 5. else go to step 10.
5. check if SNR is above threshold. If it is, go to step 6. else go to step 10
6. check if channel to be used is available. If it is go to step 7. else go to step 10
7. check if ale level is available to establish required link. If it is go to step 8. else go to step 10.
8. update Level Map with Ale level and channel. Also update channel availability
9. send ale message to the ale level through the aleMessage port.
10. go back to passive state.

4.3.7.3. Channel Availability:

It is important to note how the channel availability is taken care of in these models. When the coordinator checks if a channel is in use, it basically checks its local database, that is, it checks if any of its levels is involved in a connection on that channel. If none of the levels is, it considers the channel as free. If it has to send out a connection request that is, act as a transmitter, it sends out the message to an available ALE level which in turn sends out the message to the destination without any more checks regarding channel availability. In actual operation, the transmitter checks if the channel is already in use. If it is in use, it refrains from sending out messages on that channel in order to avoid interference. However, this is not a problem in the DEVS models because the receiver is able to check for channel use. If the channel *is already in use*, it just discards the message without affecting the transmissions already occurring on the channel.

In section 4.3.10, a global channel coordinator is introduced which prevents multiple transmissions taking place on the same channel simultaneously.

4.3.7.4. Level Map:

The coordinator maintains a hash table called the Level Map. This map consists of the different ALEs involved in connections. It also contains the stations and channels involved in these connections. Thus the key of the hash table is the station which is at the other end of the connection. The value of this key is the level, channel pair which is involved in the connection with that particular station.

When the coordinator receives a connection request (from either its station or an ALE), it looks for a free ALE level provided other conditions are satisfied. The first available level is chosen for connection and the station is entered into the Level Map as the key. The level chosen for the connection and the channel used are entered as values for this key. Whenever a level becomes available again, either because the connection terminated or could not be established in the first place, the coordinator updates its level map. Figure 33 shows the Level Map of some ALE coordinator during a simulation.

```
Printing Level Map for ALE: 1
ALE: 151        Level: 2      Channel: 3
ALE: 150        Level: 1      Channel: 1
```

Figure 33. ALE Level Map

<u>4.3.7.5. Connection Request Selection:</u>

When the coordinator receives more than one connection request simultaneously from a number of ALEs, it first puts them all in a queue. These messages are ranked in order of the SNR. If enough ALE levels are available and these connection requests satisfy all other conditions listed in 4.3.6.2, all the requests are accepted. However, if enough levels are not available, then the messages with the best SNR are accepted.

In actual practice, this is not the case. An ALE controller keeps scanning and when it scans a channel, if there is an ALE connection request on that channel, it checks for the SNR of that signal. If the SNR is above the threshold, it accepts the connection. There is no check to see if there are other requests on other channels and which one is the strongest and so on. However, in the DEVS abstract models developed in this research, the scanning phase is not modeled and the approach outlined above is treated as a fair approach. It does make sense to accept stronger signals to yield higher message throughputs since the probability of error in the transmission is less.

4.3.8. ALE Level

The ALE Level model is responsible for exchanging the three way handshake of the ALE protocol, maintaining and terminating links and the transmission of messages between the end stations. The structure of an ALE Level is shown in figure 34.

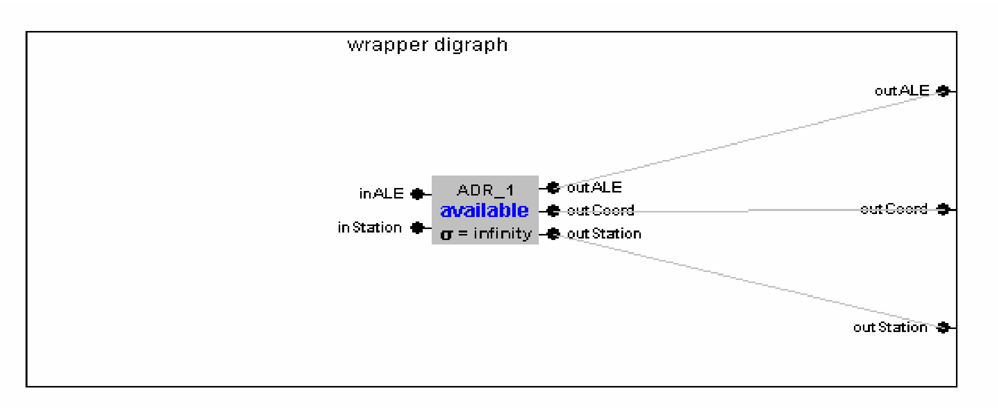

Figure 34. ALE Level

It has two input ports, “inALE” and “inStation” and three output ports “outALE”, “outstation” and “outCoord”. The ALE Level receives station messages through its inStation port and ALE messages through its inALE port. It sends messages meant for its station through the outstation port, messages meant for other ALEs through the outALE port. It sends messages to the coordinator about its availability through the outCoord port.

Each ALE level can work either as a transmitter or a receiver, although, on any one connection, it can work as only one of these.

4.3.8.1. Transmitter:

As a transmitter, the working of the ALE level is simple. It has to implement the handshake protocol and transmit messages. In order to accomplish this, the model uses a set of timers mentioned below:

- Calling Time: This is the time duration for which the model sends the connection request message to ensure that the destination ALE has a chance to receive it during its scan mode. This is set by default to 14.112 seconds.
- Wait for Response Time: Time for which the model waits after completing the call to receive a response from the receiver. Set by default to 2.744 seconds
- Acknowledging Connection Time: Time taken to acknowledge a response sent by the receiver ALE. Set by default to 2.7 seconds.
- Wait for Activity Time: Once a connection is established, it expects its station to send it a message within this time. Otherwise, it breaks the link due to inactivity. Set by default to 30 seconds

All timers can be set by the user to different values.

The model also has a rate of transmission parameter. This is used to calculate the time taken by the model to output a message completely which is given by (message size / rate of transmission). The transmission rate is set by default to 300 bps. This also can be set to any value by the user.

The algorithm for the working of the transmitter is outlined below:

1. *Start in the available phase. Stay until message is received from station*
2. *if (message received from station) go to step 3*
3. *if (message is connection request) go to step 4 else go to step 11*

4. *send ale connection request message to destination (this message contains the destination id, channel to use and the SNR value)*
5. *wait for response from channel for t1* = (CallingTime + WaitforResponseTime)
6. *if (response received within t1) go to step 7 else go to step 18*
7. *wait for* AcknowledgingConnectionTime *and send acknowledgement message*
8. *send update to station and enter connected phase*
9. *wait for t2* = WaitforActivityTime *to receive message from station*
10. *if (message received from station within t2) go to 11 else go to 14.*
11. *if (message is disconnect message and if in connected phase) go to 14 else go to 16.*
12. *if (message is not disconnect message and in connected phase) wait for t3* = (message size/ rate of transmission time) *and send message to destination*
13. *go to step 9.*
14. *send disconnect message to destination*
15. *send update message to coordinator and go to available phase*
16. *if (message is the station message to be sent to destination and in connected phase), go to 11 else go to 17.*
17. *discard message and stay in previous phase.*
18. *terminate connection attempt. Update station and coordinator. Go back to available phase.*

A point worth noting is that the transmitter sends the call and then waits for calling time plus the wait for response time. In the actual protocol, the transmitter stays in the calling phase for the calling time and then enters the wait for response time. Thus the total time since the beginning of the call to the time it receives a response could be a maximum of the calling time plus the wait for response time. The algorithm mentioned above achieves the same.

<u>4.3.8.2. Receiver:</u>

The working of the ALE level as a receiver too is simple – follow the ALE protocol. The model uses a few timers when functioning as a receiver:

- Responding Time: Time taken for the model to receive a call and send a response to the transmitter.

In addition to this, the model also uses the Wait for Response and Wait for Activity timers mentioned previously. The algorithm for the receiver is outlined below:

1. *Start in the available phase. Stay in this phase until message is received from an ALE.*
2. *if (message is connection request) go to 3. else go to 10*
3. *wait for t1 =* RespondingTime *and send connection acceptance (response) to transmitter.*
4. *wait for t2 =* WaitForResponse *time to receive acknowledgement*
5. *if (acknowledgement received from transmitter within t2) go to 6.else go to 11.*
6. *send connection update to station and enter the connected phase.*
7. *wait for t3 =* WaitForActivityTime *to receive message from transmitter*
8. *if (message received from transmitter within t3) go to 9.else go to 11.*
9. *pass on the message to the station. Go to step 7.*
10. *if (message is disconnect and in connected phase) go to 11. else go to 12*
11. *disconnect and send update to coordinator*
12. *if (message is station message from transmitter and in connected phase) go to 9. else go to 13.*
13. *discard message.*

It should be mentioned here that the entire connection process between two ALE levels in the model is ideal, that is, all connection attempts are successful. The ALE coordinator in the receiver ALE would have already checked the SNR threshold, channel and level availabilities. So, when the connection requests are passed on to the level, it means that the basic conditions have already been satisfied. Any connection rejections now are due to timing delays. These delays may arise due to different reasons. Connections may also be rejected due to large number of errors. However, if the SNR threshold test has already been passed, the probability of this occurring is rather low. Further, it should be kept in mind that the basic purpose of modeling the system is to answer higher level questions about system capacity, effect of station addition, effect of channel addition, etc. In the case of a single ALE, the more relevant question is the effect of adding levels to that particular ALE. This question can be answered easily if the calls are assumed or modeled to be rejected due to level unavailability and not due to timing delays. Although that would be an ideal case or system, it would help us in answering the higher level questions fairly accurately. At the very least, these can be considered as very abstract models and could be extended later on to incorporate additional details.

Further, when a transmitter station learns that it is connected to the receiver, it always sends a message to its ALE to transmit to the destination. Usually it sends only one message per call.

4.3.9. ALE Transducer

The ALE Transducer is responsible for collecting the ALE usage statistics of the ALE of which it is a part. The structure of an ALE transducer model is shown in figure 35. It has three input ports "inALE", "inStation" and "inLevel" and one output port "out". The transducer receives messages from its station through the inStation port, messages from other ALEs through the inALE port and messages from the different ALE Levels through the inLevel port. It sends any output through its out port.

At present, the performance measures collected at each transducer are:

- Number of Calls Attempted: This is the number of calls that the ALE sends out as a transmitter. These are the connection requests.
- Number of Calls Connected: Connected calls refer to the calls that were accepted by the receiver.

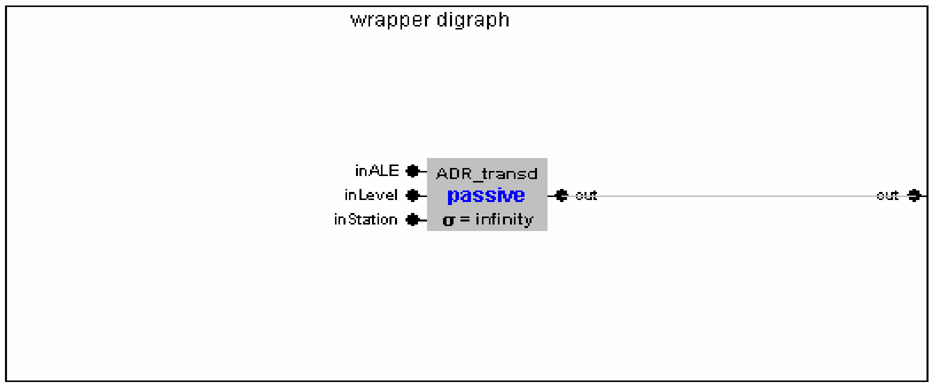

Figure 35. ALE Transducer

- Number of Calls Acknowledged: Acknowledge calls are all the connected calls which are acknowledged by the ALE Level.
- Number of Messages Sent: This is the number of messages sent by the transmitter ALE to a receiver ALE.
- Number of Calls Received: Received calls refer to the connection requests that were received by the ALE.
- Number of Calls Accepted: Accepted calls are the received calls that could be connected.
- Number of Calls Linked: Linked calls are the accepted calls that were acknowledged by the transmitter.
- Number of Messages Received: This is the number of messages that the ALE Levels received while they functioned as receivers on connections.

All these measurements are collected over the time period of the complete simulation. This time period is set by the experimenter.

It can easily be seen that the first four parameters refer to transmitter statistics and the latter four refer to receiver usage. Since this is an ideal system as stated earlier, number of accepted calls = number of acknowledged calls = number of messages sent = number of messages received. And number of accepted calls = number of connected calls. Thus, the transducers basically measure the number of calls sent out by transmitters and the number of calls that were accepted by receivers.

Each ALE Transducer reports its usage measures to the System Transducer.

4.3.10. Channel Coordinator

The Channel Coordinator is responsible for ensuring that when two signals arrive simultaneously on the same channel, their interference is taken into account and only the strongest signal is delivered to the required station. The Channel Coordinator model is shown in figure 36.

It has one input port “inALE” and one output port “outALE”. In the experiments conducted in this research, the channel coordinator is a part of the fixed station ALE set. All ALE messages sent to the fixed stations are first received by the channel coordinator through its inALE port. They are sent to the fixed station ALEs

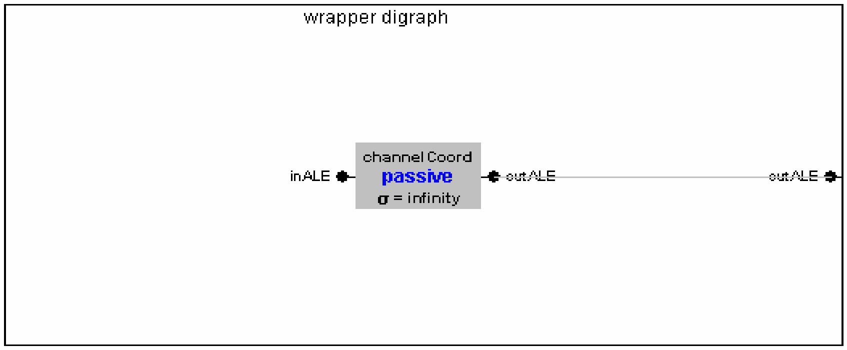

Figure 36. Channel Coordinator

through the outALE port. The channel coordinator essentially comes into play when more than one connection request is sent to ALEs on the same channel simultaneously which in actual practice may occur during a single scan cycle. The channel coordinator then examines all the signals on the same channel. The one with the strongest SNR is considered as a signal and all others are considered as interference or noise. The SNR of the strongest signal is then reduced by the noise power and is sent to the fixed station ALE.

The algorithm for the channel coordinator is presented below.

1. *Start in the passive phase. Stay in that phase until message is received.*
2. *if (message received) go to step 3.*
3. *if(message is connection request) go to 4 else go to 12*
4. *for every channel used in the system repeat step 5*
5. *if (more than one connection request received at the same time) go to 6.else go to 11.*
6. *find signal with strongest SNR. Get its signal power.*
7. *get signal powers of all other signals. Add them up as noise*
8. *calculate new SNR for the strongest signal.*
9. *pass this message on to the fixed station ALE system*
10. *if all channels examined, go to 11. else go to 4.*
11. *if only one connection request message go to 9.*
12. *pass on message to fixed ALE system.*

13. return to passive phase.

Thus, when there is more than one connection request on the same channel, the channel coordinator ensures that only one of them goes through irrespective of the destination stations. This in turn ensures that amongst all the connections in the system, there is at the most only one connection on any channel.

If a connection already exists on a channel, the channel coordinator is not able to recognize that the channel is already in use, since it does not maintain a channel database. This situation does not arise in the experiments since all connection requests are sent at the same time.

4.3.11. System Transducer

The system transducer is part of the experimental frame. It receives the different measurements from every ALE in the system and aggregates them. The System Transducer model is shown in figure 37.

It has one input port “in” through which it receives updates from the ALE Transducers and has one output port “out” through which it can send out any messages.

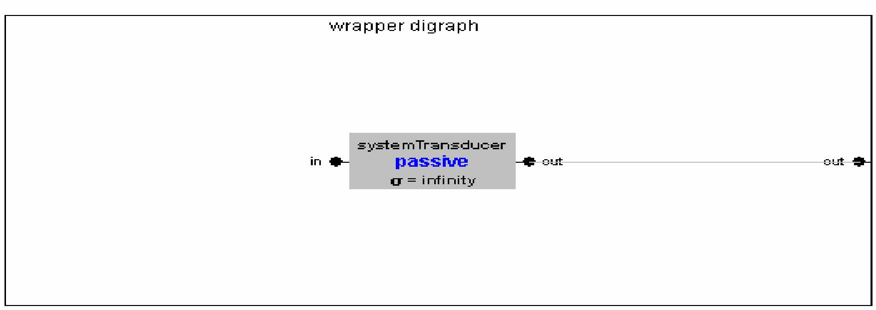

Figure 37. System Transducer

These are some of the basic components or building blocks of the system. These are interconnected to form bigger components or bigger blocks in the system.

4.3.12. ALE

An ALE is a DEVS coupled model and is made up of:

- A coordinator
- ALE Levels
- An ALE Transducer

The structure, that is, the components and the couplings, is the same for both mobile and ground station ALEs. The only difference is that mobile ALEs usually have only one ALE Level whereas the ground station ALEs would have more. Figure 38 shows a mobile station ALE and figure 39 shows a fixed station ALE with two levels.

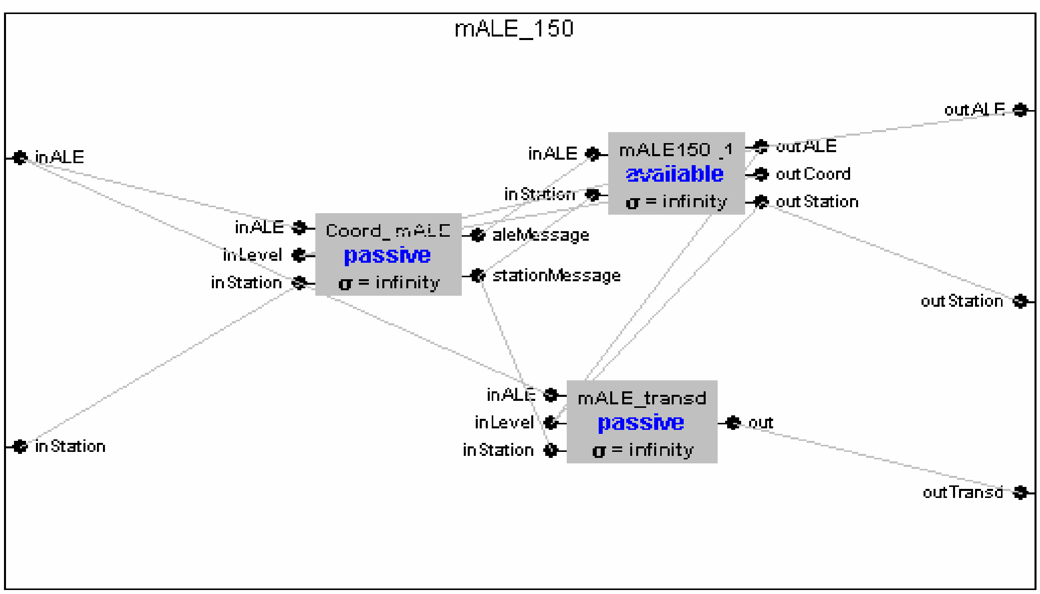

Figure 38. Mobile Station ALE

The ALE itself has two input ports "inStation" and "inALE" and two output ports "outstation" and "outALE". These ports are the ALE's interface with its station and other ALEs in the system. All messages received through its input ports are sent to the coordinator which then suitably re-directs the messages to the ALE levels. The stationMessage port of the coordinator is coupled to the inStation port of the levels. Since all station messages that have to be passed on to the levels are sent through the outstation port, the levels received station messages through the inStation port. All the other couplings are self evident and satisfy all requirements mentioned in the previous sections. Hence they are not elaborated any further.

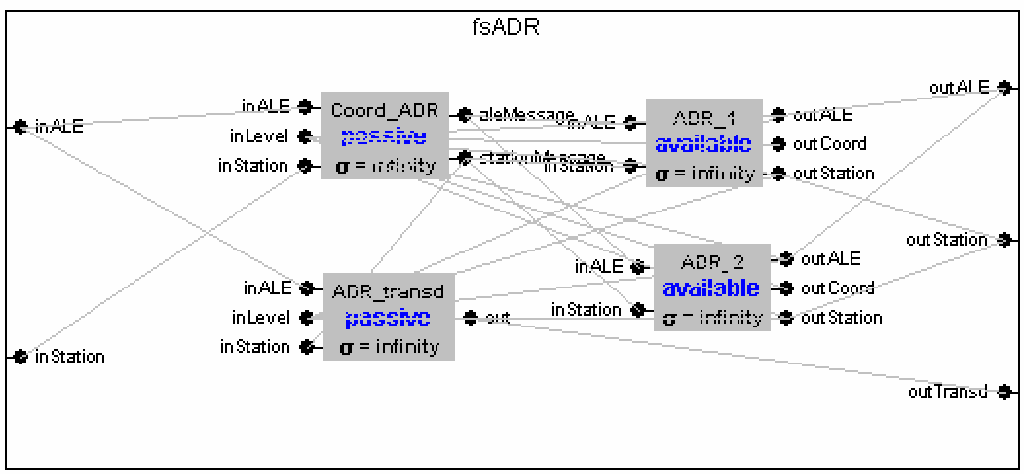

Figure 39. Fixed Station ALE

One important point is however worth noting. The stationMessage port is also coupled to the inStation port of the transducer. This would imply that the transducer actually measures the number of calls that the ALE levels attempt and not the number of calls that the station wanted the ALE to attempt. This difference would arise when a station tries to connect with a destination, but no ALE levels are available or the channel is already in use. In practice, this may not happen since the ALE operator or controller or station would already know if all the levels were busy. However, in the experiments conducted in this research, since the simulation is automated and messages are randomly generated, there is no way to ensure that this situation would not happen. Instead, the performance measurement is taken as indicated above.

4.3.13. Plane Digraph

The Plane Digraph sometimes also referred to as Aircraft Digraph is the set of planes or aircraft in the system. The Aircraft Digraph is a coupled DEVS model. An Aircraft digraph with five planes is shown in figure 40.

It acts as a wrapper around the different aircraft in the system and allows them to communicate with their ALEs through the inALE and outALE ports.

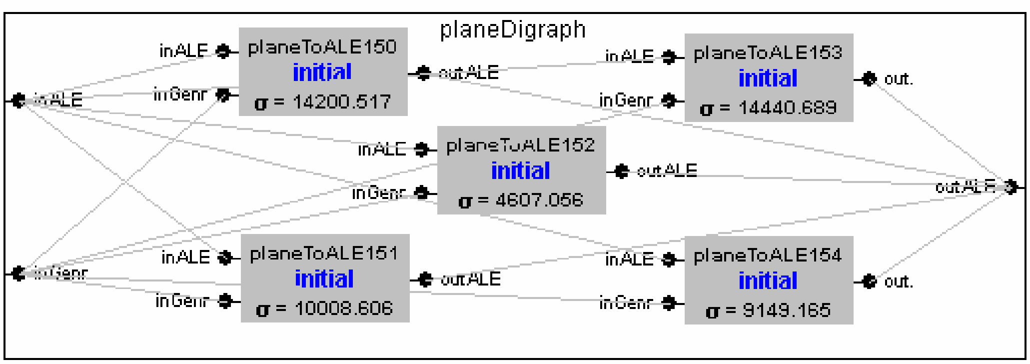

Figure 40. Plane Digraph with five planes

The number of aircraft in the system can be specified in two ways as already/ discussed in sections 4.3.3 and 4.3.4. For a large number of planes, the user usually enters the number of planes and the flight path is randomly generated. If the user enters all the plane information in a configuration file, then all information is taken from that configuration file. Each aircraft is also the same as a mobile station and has its own ID.

4.3.14. Generator Digraph

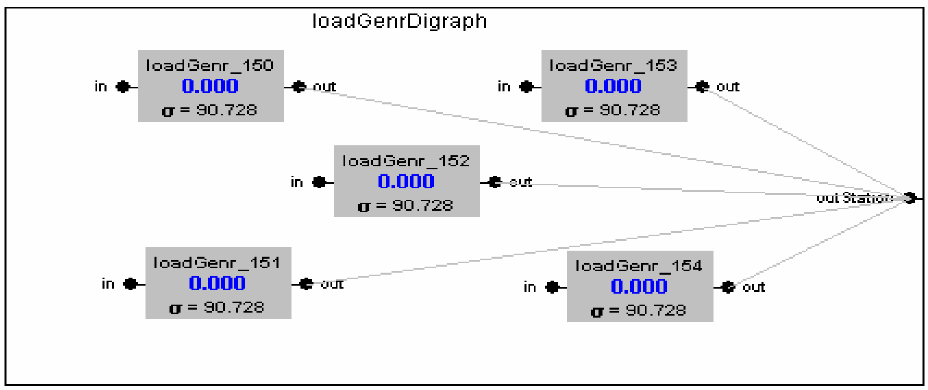

Figure 41. Generator Digraph with five generators

The Generator Digraph is a coupled model consisting of the various generators in the system. For every plane in the system, a corresponding generator is created which triggers the plane to send a message. Each generator is given the start of simulation, end of simulation and mean inter arrival time of messages. So, each generator then starts at the start of simulation and triggers the corresponding plane as decided by the exponential random variable. When the end of simulation time is reached, no more triggers are sent.

A generator digraph with five generators is shown in figure 41.

4.3.15. Mobile ALE System

The Mobile ALE System is the set of mobile ALEs in the system. There is a one-to-one correspondence between the number of planes and the number of mobile ALEs in the system since each plane has one mobile ALE. Hence, the creation of the mobile ALE set is done in the same way as the planes are created. If the user enters the number of planes in the system, this number is also taken as the number of mobile ALEs and the corresponding mobile ALEs are created. The mobile ALEs have the same IDs as their corresponding mobile station.

A mobile ALE system with five mobile ALEs is shown in figure 42.

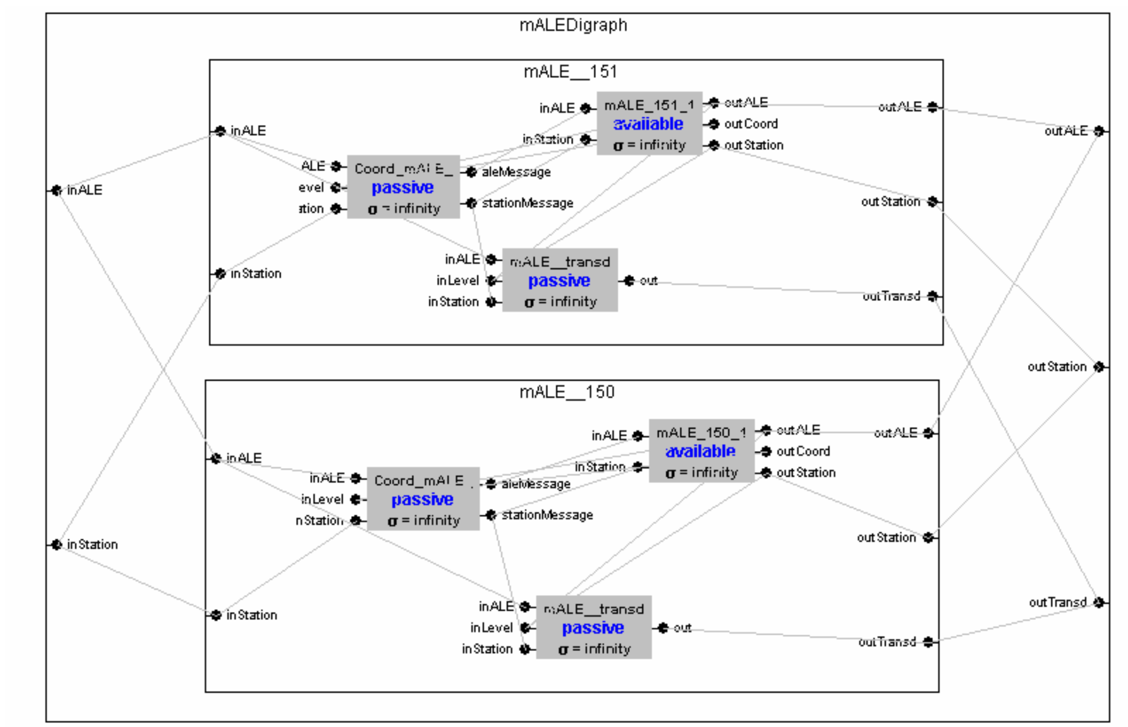

Figure 42. A Mobile ALE Digraph

4.3.16. Fixed ALE System

The Fixed ALE System consists of all the fixed station ALEs in the system and the channel coordinator. A Fixed ALE system with two ALEs and the channel coordinator is shown in figure 43.

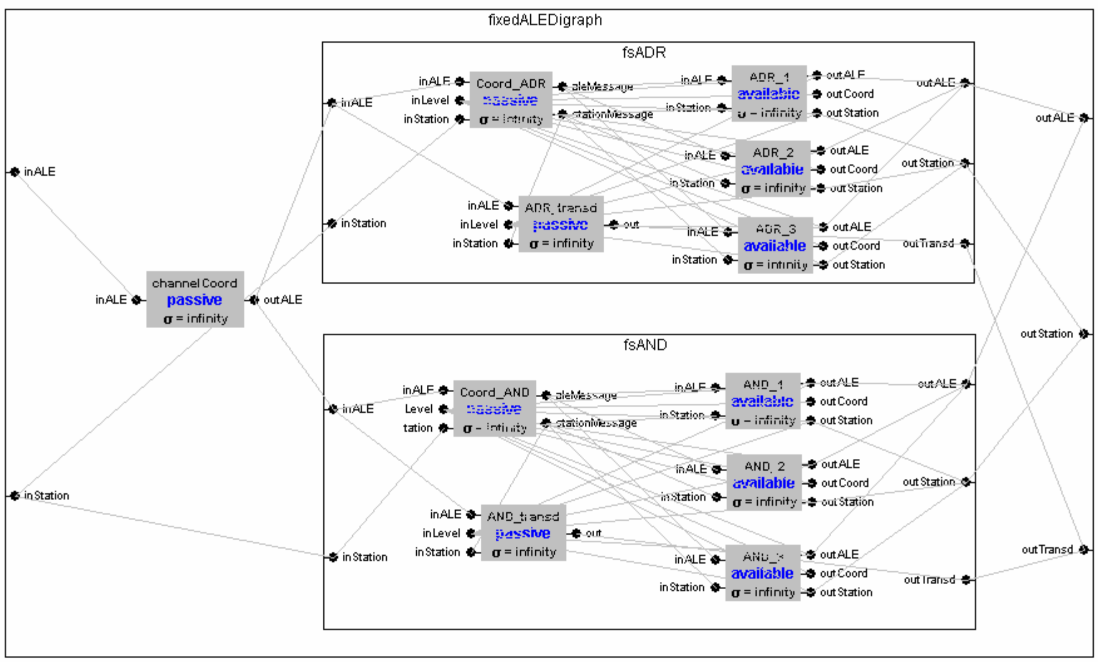

Figure 43. A Fixed ALE Digraph

The set of fixed stations is entered in configuration file which is read and corresponding fixed station ALEs are created in the system model. However, the number of levels at each fixed ALE is the same and is entered by the user. In the experiments conducted, it is assumed that all ALEs have the same number of levels.

4.3.17. ALE System

The ALE System is a coupled model consisting of the Mobile ALE System and Fixed ALE System. In terms of the architecture discussed previously, it constitutes the ALE Controller System. In DEVS terminology, it is the system under study or system under test and is sometimes referred to as the system. An ALE System with two mobile and two fixed ALEs is shown in figure 44.

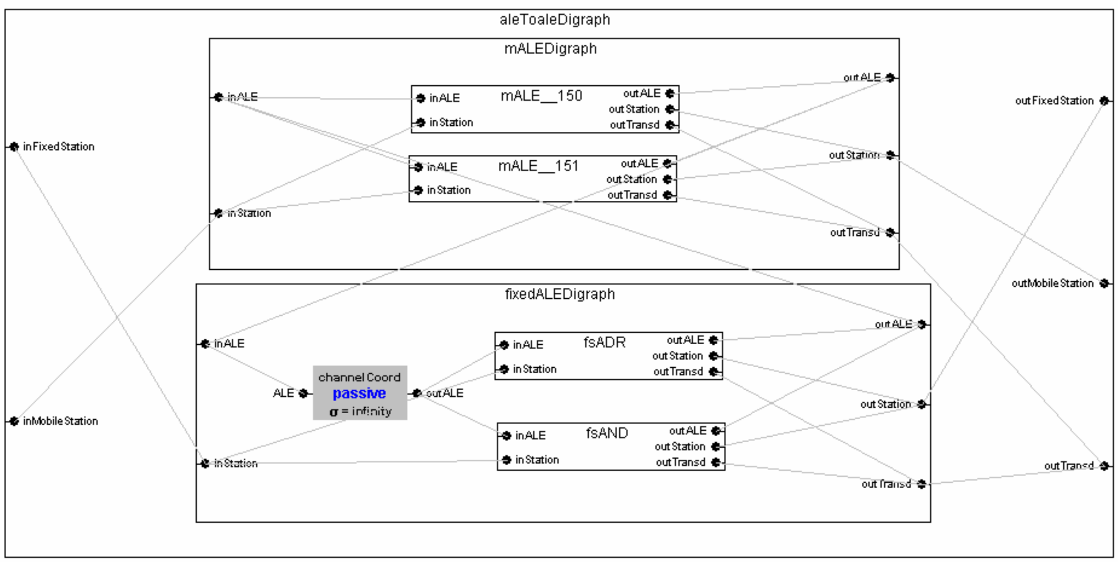

Figure 44. An ALE System

4.3.18. Experimental Frame

The Experimental Frame consists of the Generator Digraph, the Aircraft Digraph and the System Transducer. Figure 45 shows an Experimental Frame.

4.3.19. Experiment Setup

The final model required for the simulation is the Experiment Setup which consists of the Experimental Frame along with the ALE System. Figure 46 shows an example of an experiment setup with five planes and mobile ALEs and two fixed stations ADR and AND.

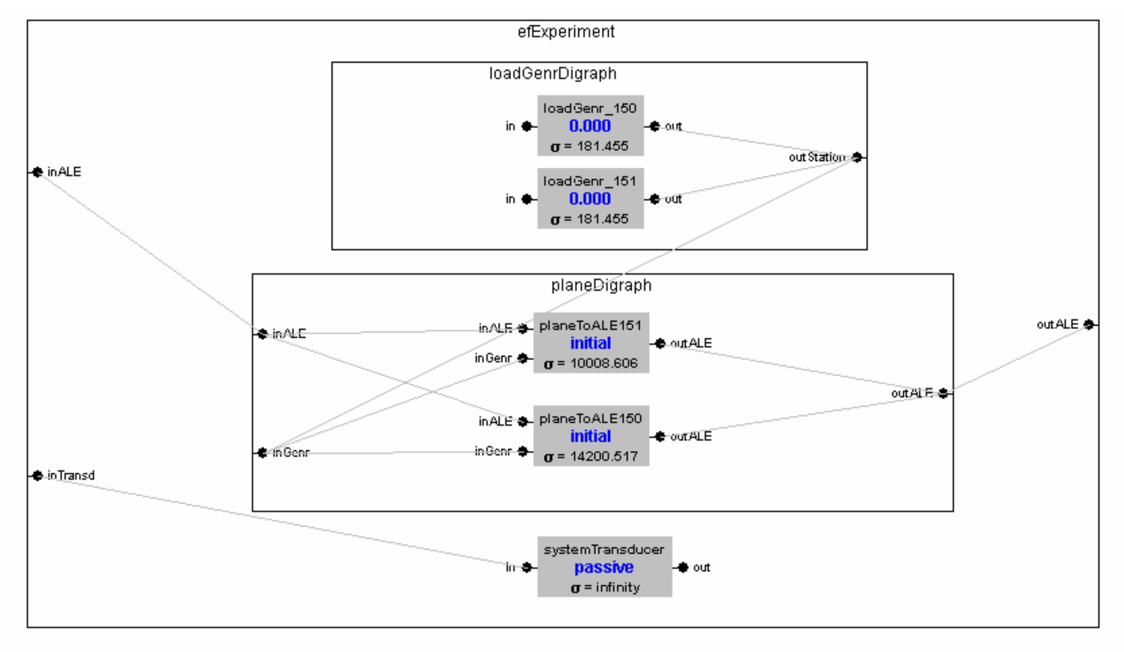

Figure 45. An Experimental Frame

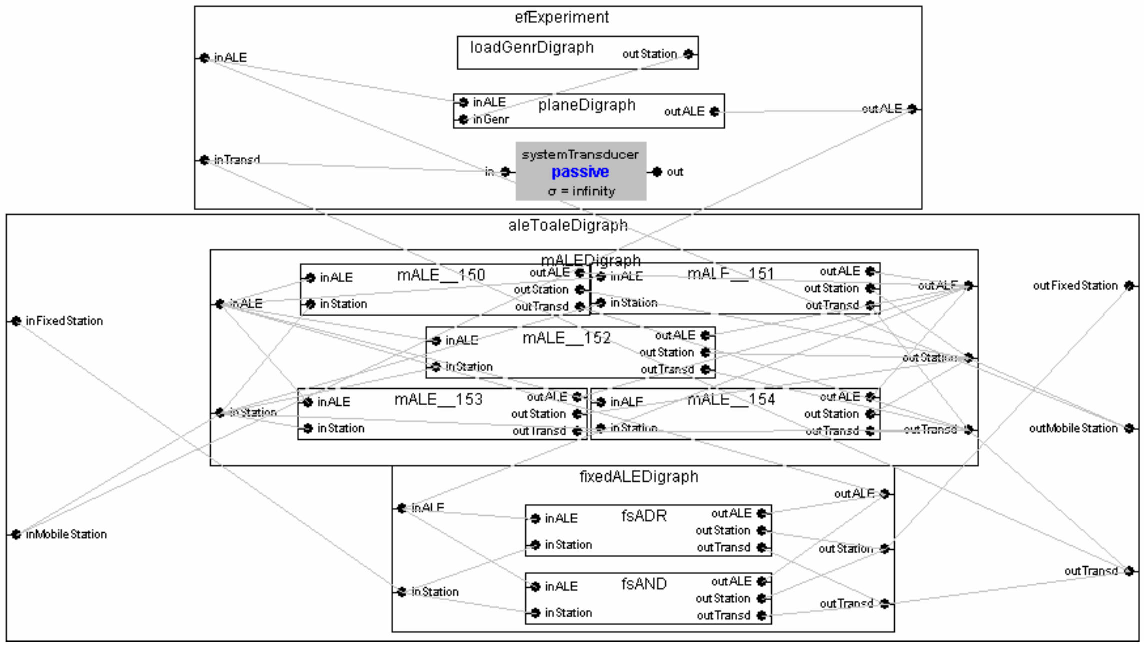

Figure 46. An Experimental Setup

4.3.20. Messages

As is evident from the above sections, the DEVS models interact with each other through explicit sending of messages from output ports to input ports.

<u>4.3.20.1. ALE Message:</u>

In this system, most of the messages exchanged are ALE messages between the different ALEs and update messages between stations and their ALEs. Hence, a new class “aleMessage” was developed.

An aleMessage has several attributes: source, destination, type, size, channel, level, transmission rate (in bps) and SNR. Since all messages exchanged in the system are such aleMessages, it is necessary to include all parameters needed in this aleMessage class. However, for certain messages, some of these parameters may not need to be used.

Source: The station ID and consequently the ALE ID of the station which initiates any connection is the source field.

- Destination: The destination station ID and consequently its ALE ID form the destination field.
- Type: Type of the aleMessage is used to distinguish between the various messages. For example, type 11 is used by a station to instruct its ALE to establish a connection with the destination specified in the destination field; type 1 is used by transmitter ALE to send a connection request to receiver ALE.
- Size: Size of the aleMessage. Used for the exchange of data messages between end stations.
- Channel: Channel used for the communication, that is, the channel on which this message is being sent.
- Level: Used within an ALE, the level field indicates the level to be used for a connection. This is used only when a connection needs to be established. This field is set by the coordinator in type 1 and 11 messages before passing on to the various levels.
- Transmission rate: The rate of transmission to be used for transmitting a message in bps. Default value is 300bps.
- SNR: The SNR value of the signal. The source of connection, which in the experiments in this thesis, is always a mobile station, sets this value after calculating it from the ICEPAC data.

4.3.20.2 Transducer Message:

Since the ALE transducers and System transducer also exchange messages, a "aleTransdMessage" class was developed. Whenever an ALE transducer updates any of its measurements, it sends an aleTransdMessage to the system transducer which then updates its corresponding measurement. The message has three fields:

- ID: Holds the station ID of the transducer sending the message
- Type: Identifies the measurement being updated. For example, type 1 signifies calls attempted, type 2 stands for calls connected and so on.
- Value: The value of the measurement to be updated. For example, if at a time t, the ALE received two connection requests, the transducer would send a type 5 message with the value set as 2.

5. Experiments and Results

In this chapter, the various experiments conducted and the results of these experiments are provided. These experiments also demonstrate how the models developed are useful in answering some of the questions raised before.

5.1. Experimental Setup

It is helpful to know various configuration details that are generally used in in this study or represent the set of configurations that could be used.

5.1.1. Waypoint List

The set of waypoints involved in all the flight routes is provided in Table 2. There are eighteen waypoints in the system. All flight routes are generated using these eighteen waypoints.

5.1.2. Fixed Station Set

Table 3 shows the set of fixed stations used in the system. There are a maximum of fourteen stations. However, not all of them will be used in every experiment. Only a subset of stations is used to demonstrate effects of adding stations, locations of stations, etc.

WAYPOINT	LATITUDE	LONGITUDE
Dover	39.133	-75.45
Croughton	51.983	-1.200
Frankfurt	50.5	8.0
Spain	41	-4
Lajes	38	-29
Salinas	18.25	-65.633
Ascension	-7.95	-14.37
Oslo	60	10.5
Stockholm	59	18
Moscow	55.5	37.5
Hungary	46.2	18.3
Riyadh	24.5	46
Vicenza	47	10.5
Spain2	41	4
Accra Ghana	5.5	0
Nairobi	1.5	37
Diego	7.333	72.367
Calcutta	22.5	88.2

Table 2. Set of Waypoints used in the Experiments

Station	Latitude	Longitude
ADR	38.817	-76.867
AND	13.583	144.933
ASC	-7.983	-14.4
CRO	51.983	-1.2
DGA	-7.333	72.417
ELM	61.267	-149.767
HIC	21.317	-157.867
INC	37.0	35.417
LAJ	38.767	-27.1
MCC	38.650	-121.383
OFF	41.267	-95.990
SAL	17.953	-66.3
THU	76.517	-68.6
YOK	35.75	139.35

Table 3. Set of Fixed Stations

5.1.3. Channel Set

Table 4 indicates the set of channels and corresponding frequencies that are used in the system.

Channel	Frequency (MHz)
1	1
2	1
3	2
4	2
5	3
6	3
7	3
8	4
9	4
10	4
11	5
12	5
13	6
14	7
15	8
16	9
17	10
18	11

Table 4. Set of channels and frequencies

5.1.4. Number of Planes, Mobile ALEs and Flight Routes

The number of planes varies from experiment to experiment and is entered by the user. Thus, since no configuration file is used, the flight routes for these planes are generated randomly, that is, the source and destination are chosen randomly. The waypoint network is created by specifying a distance of 4000 kilometers. That is, all waypoints separated by a maximum great circle distance of 4000 km are connected to each other. All flight routes are the shortest path between the source and destination waypoints. An additional restriction placed on the selection of source and destination is that they must have at least one hop between them. In other words, the flight route should be made up of at least two flight segments. This is to ensure that the planes are mobile for at least some reasonable amount of time during a simulation. Since the number of planes is created dynamically, corresponding mobile ALEs in the ALE system are also created dynamically.

Thus, various experiments could be generated by changing just the number of planes.

5.1.5. Simulation and Message Generation Times

All simulations start at time 0 seconds and run for 9 hours (32400 seconds). After 9 hours no more messages will be generated in the system. However, if necessary, the planes will continue to move and reach their destinations. The simulation time is basically meant to control the generators.

One generator is created for every plane in the system. All messages are given the same start of simulation and end of simulation times. Further, they are given the same mean message generation times. This would make them trigger their respective planes at the same time. This would involve a scenario in which there would be a maximum probability of collisions between calls. The number of connected calls might be affected due to this nature of message generation. However, since the results are meant to be indicative and not actual suggestions, this scenario was not considered as wrong.

In all experiments involving different configurations, three different mean generation times are used: 10 minutes (600 seconds), 20 minutes (1200 seconds) and 30 minutes (1800 seconds). These three values were considered representative of high traffic, medium traffic and low traffic in that order.

5.1.6. Month and Sunspot number

As has been mentioned earlier, since the ionosphere structure changes from day to day, hour to hour and year to year, some of these parameters need to be fixed for all experiments. The month is set as June and the sunspot number is always set as 100.

5.2. Experiment Methodology

The various experiments conducted serve to observe the effects of SNR, addition of levels, addition of stations, location of stations and addition of channels on the performance indices described in section 5.2.5.

5.2.1. SNR

These experiments demonstrate the effect of the SNR threshold or in other words the effect of propagation conditions on the system performance. This is seen by first setting the SNR threshold to 0 dbW (decibel Watts). This would mean that all calls pass the SNR test. And that is precisely the effect of good propagation conditions – allowing all signals to preserve an SNR based on the transmitted power and interference from other signals with very little noise added by the channel itself.

SNR threshold is then increased to 200dbW. With this threshold, not all calls would pass the SNR test. This would be a way of introducing more interference from the channel thus effectively reducing a signal's SNR and its chances of passing the threshold test.

In these experiments, the number of planes, their flight routes, the number of fixed stations, the number of levels in each station and the number of channels that can be used in the system are kept constant. The number of calls sent by the transmitter and the number of calls accepted by the receiver are measured in each case.

5.2.2. Addition of Levels

In these experiments, the number of planes, their flight routes, the number of fixed stations and the number of channels that can be used in the system are kept constant. The SNR threshold is set to 0. With this constant fixture, the number of levels is varied. We may note that since the number of levels at each station is the same, an increase in the number of levels implies that all stations experience the same amount of increase in different experiments.

5.2.3. Addition of Stations

In these experiments, the number of planes, their flight routes, the number of levels at each station and the number of channels that can be used in the system are kept constant. The SNR threshold is set to 0. With this constant fixture, stations are added to the system. Two questions can be answered using such experiments:

- Effect of adding stations

- Effect of adding specific stations – if certain stations, say, which are close to the theater of operations are added, how much of an improvement is seen in the system.

5.2.4. Addition of Channels

In these experiments, the number of planes, their flight routes, the number of levels at each station and the number of stations in the system are kept constant. The SNR threshold is set to 0. Then the system is allowed to use different number of channels and performance measurements are analyzed.

5.2.5. Performance Indices

The main performance indices are the number of calls attempted by the mobile station ALEs and the number of calls accepted by the ground station ALEs. While the former remains the same for a given configuration of planes and mean generation time, the latter would depend on the factors discussed above. Hence, it would serve as a good parameter to evaluate system performance.

Although other measurements such as calls acknowledged, messages sent and received and others are collected, since this is an ideal as discussed earlier, they all turn out to be equal to the number of calls connected.

5.3. Experiments and Results

In the tables outlining the results in the following sections, the abbreviations used stand for

CA - number of calls attempted,

CC - number of calls connected,

CACK - number of calls acknowledged,

MS - number of messages sent

CREC – number of calls received

CACC – number of calls accepted

CL – number of calls linked

MR – number of messages received.

These measurements are taken from the system transducer and hence, are a measure of the aggregate system performance.

5.3.1. Determine SNR Effects

Experiment 1:
Planes = 1
FS = 1
Levels = 1
SNR Threshold = 0

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	51	51	51	51	51	51	51	51
20	26	26	26	26	26	26	26	26
30	16	16	16	16	16	16	16	16

<u>Experiment 2:</u>
Planes = 1
FS = 1
Levels = 1
SNR Threshold = 200

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	51	31	31	31	51	31	31	31
20	26	17	17	17	26	17	17	17
30	16	7	7	7	16	7	7	7

In experiment 1, an ideal system of one plane, one station with one level and 0 SNR threshold is simulated. As expected all the calls are accepted. When the threshold is increased to 200 in experiment 2, the number of calls accepted is lower than the first experiment for all three message generation rates. When the mean message generation time is 10 minutes, the number of calls accepted falls to 65.38% in experiment 2 compared to 100% in experiment 1. When the mean message generation time is 20 minutes, the number of calls accepted falls to 60.78% and when it is 30 minutes, the number of calls accepted falls to 43.75%. Thus, deterioration in system performance is seen in all the three cases in experiment 2.

<u>Experiment 3:</u>
Planes = 2
FS = 1
Levels = 1
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	102	51	51	51	102	51	51	51
20	52	26	26	26	52	26	26	26
30	32	16	16	16	32	16	16	16

<u>Experiment 4:</u>
Planes = 2
FS = 1
Levels = 1
SNR Threshold = 200

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	102	43	43	43	102	43	43	43
20	52	21	21	21	52	21	21	21
30	32	13	13	13	32	13	13	13

In experiments 3 and 4, there are two planes in the system. Therefore even with ideal 0 threshold case, not all calls should be accepted since there is only one level in the fixed station. When the threshold is increased, the number of calls accepted drops even further. In experiment 3, on an average, 50% of the calls are accepted. In experiment 4, on an average only about 41% of the calls are accepted.

<u>Experiment 5:</u>
Planes = 2
FS = 1
Levels = 2
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	102	93	93	93	102	93	93	93
20	52	48	48	48	52	48	48	48
30	32	30	30	30	32	30	30	30

<u>Experiment 6:</u>
Planes = 2
FS = 1
Levels = 2
SNR Threshold = 200

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	102	68	68	68	102	68	68	68
20	52	34	34	34	52	34	34	34
30	32	16	16	16	32	16	16	16

In experiments 5 and 6, two levels are provided in the fixed station. However, even in the 0 threshold case, not all calls were accepted. This is due to channel availability or rather an attempt to use the same channel in which case only one of the calls goes through. In the other case, the number of calls accepted drops down even more since the SNR is not strong enough. The number of channels available is the same in both cases. Hence it is the SNR which causes the drop in the number of calls accepted. The effects of the number of available channels available are explored in section 5.3.3.

Thus, the propagation conditions modeled as an increase/decrease in the SNR threshold give a correct picture of the system performance. To get accurate measurements, this increase/decrease in the threshold will have to be modeled correctly.

5.3.2. Addition of levels/Addition of stations:

Experiment 7:
Planes = 5
FS = 2
Levels = 2
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	255	161	161	161	255	161	161	161
20	130	86	86	86	130	86	86	86
30	80	52	52	52	80	52	52	52

Experiment 8:
Planes = 5
FS = 2
Levels = 3
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	255	190	190	190	255	190	190	190
20	130	101	101	101	130	101	101	101
30	80	58	58	58	80	58	58	58

In experiment 7, with five planes and two fixed stations with two levels each and a threshold of 0, around 65% of the calls were accepted. When the number of levels was increased to 3, the calls accepted increased to an average of 77%. Thus, adding levels for this particular configuration which includes the flight route of the planes was a good move.

Experiment 9:
Planes = 5
FS = 3
Levels = 2
SNR Threshold = 0

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	255	183	183	183	255	183	183	183
20	130	100	100	100	130	100	100	100
30	80	58	58	58	80	58	58	58

In experiment 9, the same configuration as experiment 7 was chosen but an additional station was added with all the stations having two levels. The percentage of calls accepted increased to an average of 73% from the configuration in experiment 7.

Experiment 10:
Planes = 5
FS = 3
Levels = 2
SNR Threshold = 0

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	255	180	180	180	255	180	180	180
20	130	101	101	101	130	101	101	101
30	80	56	56	56	80	56	56	56

In experiment 10, a different station was chosen at random to be the third station. Calls accepted increased to an average of 73%. This is the same result as in experiment 9, which implies that either of the stations could be added and the performance improvement would be the same.

However, adding a level to each of the stations (experiment 8) yielded a better performance increase. So in this case, that would be a better solution.

<u>Experiment 11:</u>
Planes = 5
FS = 3
Levels = 3
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	255	180	180	180	255	180	180	180
20	130	111	111	111	130	111	111	111
30	80	59	59	59	80	59	59	59

In experiment 11, the same set of fixed stations as experiment 9 was chosen, but each station had three levels. The calls accepted increased to nearly 75% which is a better performance than the performance in experiments 9 or 10. But this solution would involve adding a station and a level in each station. The choice between these two solutions can only be decided on a cost/performance analysis which is up to the designer.

<u>Experiment 12:</u>
Planes = 15
FS = 3
Levels = 3
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	765	232	232	232	765	232	232	232
20	390	129	129	129	390	129	129	129
30	240	79	79	79	240	79	79	79

<u>Experiment 13:</u>
Planes = 15
FS = 3
Levels = 7
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	765	243	243	243	765	243	243	243
20	390	138	138	138	390	138	138	138
30	240	71	71	71	240	71	71	71

<u>Experiment 14:</u>
Planes = 15
FS = 7
Levels = 3
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	765	229	229	229	765	229	229	229
20	390	131	131	131	390	131	131	131
30	240	75	75	75	240	75	75	75

<u>Experiment 15:</u>
Planes = 15
FS = 7
Levels = 7
SNR Threshold = 0

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	765	230	230	230	765	230	230	230
20	390	133	133	133	390	133	133	133
30	240	75	75	75	240	75	75	75

Experiments 12 to 15 explore pretty much the same kind of effects as in experiments 7 to 9. Here there are 15 planes. Initially there are three stations with three levels each. Initially levels are added, then stations are added and finally both stations and levels are added. The percentage of calls accepted in each case is between 31% and 32%. Hence, none of the three solutions seems to work particularly well.

5.3.3. Addition of Channels

In all the experiments conducted above, the full set of channels was used. Since data for additional channels (ICEPAC) data was not available, the effect of number of channels in the system was studied by decreasing the number of channels in the system.

Experiment 16:
Planes = 15
FS = 7
Levels = 7
SNR Threshold = 0
No of channels = 10

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	765	133	133	133	765	133	133	133
20	390	69	69	69	390	69	69	69
30	240	43	43	43	240	43	43	43

In experiment 16, the setup is the same as experiment 15. But the number of channels available now is only 10. As can be seen, the percentage of calls accepted drops to around 18% from about 31% in experiment 15.

Experiment 17:
Planes = 5
FS = 3
Levels = 3
SNR Threshold = 0
No of channels = 10

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	255	145	145	145	255	145	145	145
20	130	79	79	79	130	79	79	79

30	80	44	44	44	80	44	44	44

In experiment 17, the configuration is the same as experiment as experiment 11, but the number of available channels is only 10. Only about 55% of calls are accepted as an average.

5.4. Basic Statistical Analysis

In this section, an example of how the experimental results can be used to perform a basic statistical analysis is illustrated. The procedure used to perform this analysis is as follows:

Two experimental configurations similar to the ones seen in the previous section are chosen. Configuration 1 is first considered. An initial seed is chosen and the simulation is run with different message inter arrival times. This initial seed is used to initialize the random generator for the first aircraft in the configuration. Each subsequent aircraft is initialized with a fixed increment to the previous seed. Thus if there are three aircraft in the system, the initial seed is 1000 and the fixed increment is 500, 1000 is the seed for the random generator of aircraft 1, 1500 is the seed for aircraft 2 and 2000 is the seed for aircraft 3. This is repeated with different values of the initial seed. After all the different simulations are run, the mean number of calls accepted for all the seed values and the standard deviation are calculated for each message inter arrival time. The same experiments are conducted against the second configuration and the mean and standard deviation are calculated.

To compare the performance of the two configurations or systems, the mean values can be used. For example, if the mean value for the first configuration is higher, then, that configuration is obviously the better one to use. The standard deviation can be used to compare the dispersion of the two systems, that is the value by which the actual number of calls accepted differ from the mean. The configuration with the smaller value is a more consistent system, in that, the results of different experiments are closer to the mean.

5.4.1. Experiment Configurations and Results

The configurations used to perform the basic statistical analysis and the results are described in this section. Two configurations were considered and results were generated for these two.

Configuration 1:

Planes = 30
FS = 14
Levels = 10
SNR Threshold = 200
No of channels = Full set of channels
Seed Increment = 10
Simulation Duration = 3 hours

Five different initial seeds were used and the simulations were run against all five of these values. For each seed value, simulations were run for mean message interval arrival times of 10 minutes, 20 minutes and 30 minutes. All simulations were run for three hours. The results are shown below.

Initial Seed = 1000:

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	495	403	403	403	495	403	403	403
20	236	213	213	213	236	213	213	213

30	154	142	142	142	154	142	142	142

Initial Seed = 2000:

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	532	428	428	428	532	428	428	428
20	273	238	238	238	273	238	238	238
30	179	164	164	164	179	164	164	164

Initial Seed = 3000:

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	471	398	398	398	471	398	398	398
20	227	208	208	208	227	208	208	208
30	155	146	146	146	155	146	146	146

Initial Seed = 4000:

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	550	445	445	445	550	445	445	445
20	288	257	257	257	288	257	257	257
30	186	171	171	171	186	171	171	171

Initial Seed = 5000:

Mean Time	**CA**	**CC**	**CACK**	**MS**	**CREC**	**CACC**	**CL**	**MR**
10	486	414	414	414	486	414	414	414
20	245	223	223	223	245	223	223	223
30	160	152	152	152	160	152	152	152

The mean value of the calls accepted and the standard deviation for each message inter arrival time are tabulated in the tables below.

Mean values:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	506.8	417.6	417.6	417.6	506.8	417.6	417.6	417.6
20	253.8	227.8	227.8	227.8	253.8	227.8	227.8	227.8
30	166.8	155	155	155	166.8	155	155	155

Standard Deviation:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	29.51	17.14	17.14	17.14	29.51	17.14	17.14	17.14
20	23.03	17.84	17.84	17.84	23.03	17.84	17.84	17.84
30	13.17	10.92	10.92	10.92	13.17	10.92	10.92	10.92

Configuration 2:

Planes = 30
FS = 10
Levels = 10
SNR Threshold = 200
No of channels = Full set of channels
Seed Increment = 10
Simulation Duration = 3 hours

For the second configuration too, the same seed values as those for configuration 1 were chosen and the experiments were conducted the same way. That is, the same message inter arrival times and simulation duration were chosen.

Initial Seed = 1000:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	495	397	397	397	495	397	397	397
20	236	205	205	205	236	205	205	205
30	154	142	142	142	154	142	142	142

Initial Seed = 2000:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	531	421	421	421	531	421	421	421
20	273	236	236	236	273	236	236	236
30	179	161	161	161	179	161	161	161

Initial Seed = 3000:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	471	391	391	391	471	391	391	391
20	227	205	205	205	227	205	205	205
30	155	144	144	144	155	144	144	144

Initial Seed = 4000:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	550	437	437	437	550	437	437	437
20	288	252	252	252	288	252	252	252
30	186	171	171	171	186	171	171	171

Initial Seed = 5000:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	486	403	403	403	486	403	403	403
20	245	224	224	224	245	224	224	224
30	160	148	148	148	160	148	148	148

The mean values of calls accepted and the standard deviation for each message inter arrival time are tabulated below.

Mean values:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	506.8	409.8	409.8	409.8	506.8	409.8	409.8	409.8
20	253.8	224.4	224.4	224.4	253.8	224.4	224.4	224.4
30	166.8	153.2	153.2	153.2	166.8	153.2	153.2	153.2

Standard Deviation:

Mean Time	CA	CC	CACK	MS	CREC	CACC	CL	MR
10	29.34	16.90	16.90	16.90	29.34	16.90	16.90	16.90
20	23.03	18.16	18.16	18.16	23.03	18.16	18.16	18.16
30	13.17	11.09	11.09	11.09	13.17	11.09	11.09	11.09

5.4.2. Performance Analysis

Figure 47 shows a plot of the means of the two configurations:

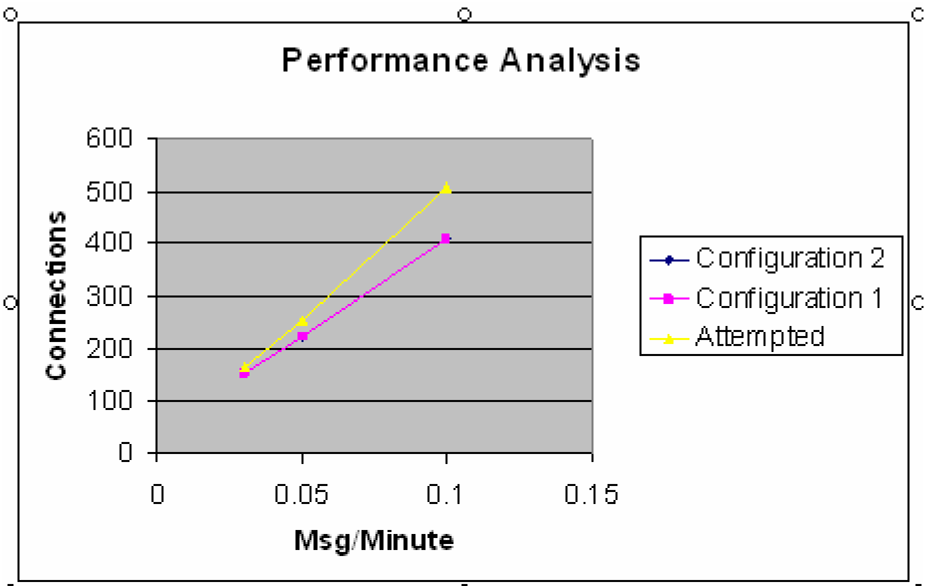

Figure 47. Plot of mean number of calls accepted

As can be seen from the graph, the mean number of calls accepted in the two configurations is nearly the same. Hence it can be concluded both configurations perform equally well under the given set of experimental conditions.

Figure 48 shows a plot of the standard deviations.

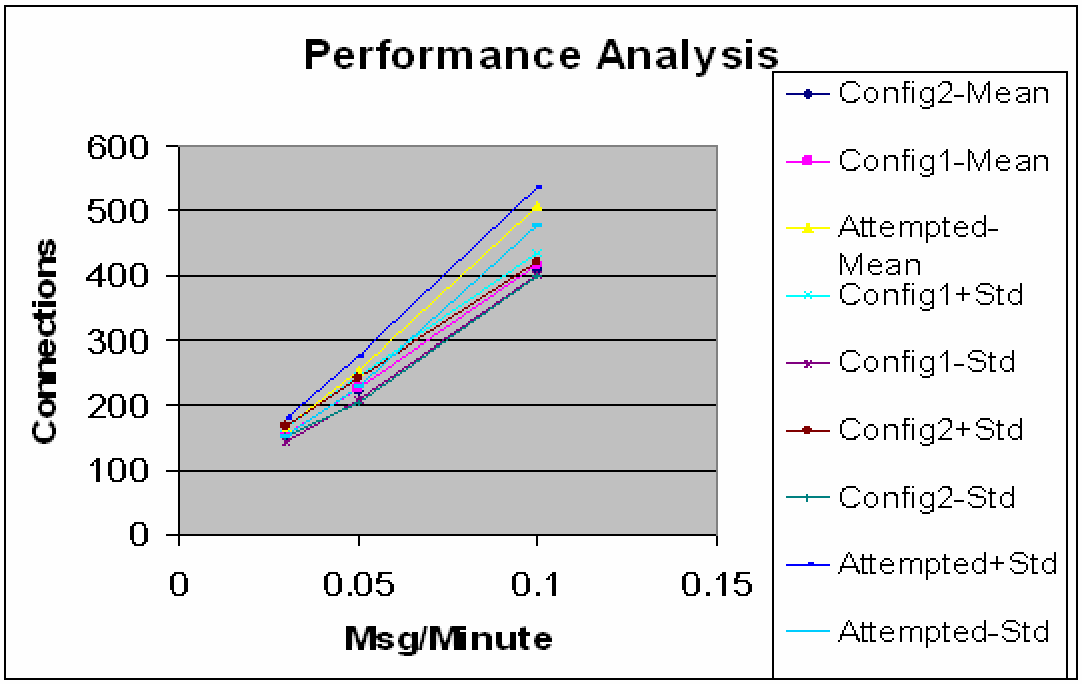

Figure 48. **Plot of the standard deviations**

The graph shows that the standard deviation too in both the configurations is nearly the same. Thus, both the configurations are equally consistent under the given set of experimental conditions.

5.5. Summary

These different experiments show how the abstract models developed using DEVS could be run in various scenarios and performance measurements would indicate how good the solutions are or what are the effects of adding those particular resources.

The experiments conducted prove that system performance improved with an addition of levels in each existing station in most cases. When the SNR of the signals is stronger, the system performance is better. Although this is an obvious fact, the experiments help in quantifying the improvement. Addition of stations however did not always lead to an improvement in system performance. The experiments help in finding out where the stations have to be added in order to improve the performance. When the number of available channels is reduced, system performance deteriorates. The experiments help in quantifying the fall in performance.

In fact, more sophisticated performance measurements can be taken, for example – the number of calls rejected at each station due to level unavailability, channel unavailability and so on. Analysis can then be conducted based on these measures and additional information could be gleaned. For example, using the measurements above, the amount of utilization of levels at each station, the amount of utilization of stations themselves can be evaluated. Then decisions can be taken as to whether the stations are being utilized enough to justify their presence or can they be removed and added elsewhere without affecting system performance.

Further the results can be used to perform a regression analysis which can help isolate the effects of each of the factors or variables affecting system performance. This can also help in identifying which factors affect system performance to a greater extent, that is, which factors are more important to consider while designing the system.

6. Conclusion and Future Work

6.1. Conclusion

In this thesis, the Discrete Event System Specification formalism was used to build abstract models of the system components and an Experimental Frame and run various simulations to study planned network and communication system expansion. The US Air Force SCOPE Command System was taken as a test network.

The SCOPE command system uses Automatic Link Establishment capable radios to set up links in the HF spectrum range. This involves propagation through the ionosphere and hence, the actual signal propagation strength depends on various factors. The ICEPAC database was used to provide approximate values of these signal strength values for different experiments. The designers of the SCOPE system face various questions such as which resources to add and how to add them to improve system performance.

A very simple network model with a few nodes and links was first built as a pilot study to study how DEVS could be used to build abstract models and whether results could be derived. The network model built and studied did indeed show that DEVS could be used for such purposes.

A new architecture for the simulation of the system was suggested. The mobility and message generation part became part of an experimental frame and the ALE protocol part was made part of the system under study. This provided the modeler with many advantages, the most important of which was the ability to develop ALE models at various resolutions independent of message generation scenarios and ANR values accesses.

Various experiments with different system configurations were then run. Although they were not replications of actual scenarios, the results obtained showed how the DEVS abstract models developed could be used in various settings to answer the questions of the SCOPE system designers. A basic statistical analysis was performed to show how performance studies of various system configurations can be conducted.

6.2. Future Work

Future work could include developing DEVS models which model all the details of the NetSim models, that is, reconvert NetSim to DEVS and run the same Experimental Frame outlined in chapter five against these models. The results from these experiments can be used to compare with the results generated with the abstract models and assess how good the abstract models are.

Future work could include performing a regression analysis which can help isolate the effects of each of the factors or variables affecting system performance. This can also help in identifying which factors affect system performance to a greater extent, that is, which factors are more important to consider while designing the system.

Future work could also involve extending the transducers developed in the system to include details about the calls were actually rejected at each ALE such as why they were rejected: were they rejected because of unavailable ALE levels, were they rejected because of weak SNR, etc. Although measurements were collected at each ALE in the system regarding how many calls were received by them and how many were accepted, they were not analyzed. Analysis of such information would lead to gaining knowledge about the amount of utilization of each station. Combined with the first suggestion, decisions can then be taken whether to increase levels at the station in question, or decrease levels at the station and also whether the station itself is being utilized enough to justify its existence. Further, if stations at some places were overloaded, stations from other less used places can be moved to these locations and the system performance can be studied.

To study the effects of propagation conditions, this study basically used the variation of SNR threshold. Another approach would have been developing a simple model in the system which would randomly affect the SNR of all signals in the system. Such a model can be placed in the experimental frame to randomly decrease each signal's SNR by some value. This value can follow a uniform distribution with good estimates of minimum and maximum values by which the SNR should be decreased.

The plane models in this study tried to connect to a station only once. In case this attempt failed, the models here did not attempt any other connections. Different station, channel selection algorithms can be used to provide alternate or the next best choices for a plane in case the first attempt failed. The performance of these choices can be studied to yield good selection algorithms.

Future work could also include developing a model of the 5066 protocol stack and including it as a factor in the system design. Simulations can then be run to find the optimal values for the protocol parameters.

A graphical display which can show the locations of the aircraft on a map can also be developed. This display should have the capacity to update the locations of the aircraft as they move along their flight route. This can provide a good visualization of the system.

These are a few of the additions to this study which can be used to yield better solutions to the SCOPE System designers. The advantage of building abstract models is that they can then be tailored to mimic any modifications the designers wish to include and study. Building a standard experimental frame will help in studying different system configurations against a standard or base set of conditions and thus comparisons between these choices can be easily performed. This thesis provides an indication as to how such studies can be performed and demonstrates the usefulness of the development of abstract models and Experimental Frame facilitated by the DEVS formalism.

REFERENCES

[1] Hild, D.R., “Discrete Event System Specification (DEVS) Distributed Object Computing (DOC) Modeling and Simulation,” PhD Dissertation, Electrical and Computer Engineering Department, University of Arizona, Tucson, 2000.

[2] SCOPE COMMAND SYSTEM SPCIFICATION – http://www.rockwellcollins.com/ecat/gs/SCOPE_Command.html?smenu=101

[3] Eric E. Johnson, " Interactions Among Ionospheric Propagation, HF Modems, and Data Protocols," Proceedings of the 2002 Ionospheric Effects Symposium, IES '02, Alexandria, VA, May 2002.

[4] Harvey, S., Skutt, M., Sanford, A., “SCOPE Command Network Simulation,” Presentation to ACIMS Research Team, August 2004.

[5] Automatic Link Establishment System - MIL-STD-188-141B, Appendix A

[6] Eric E. Johnson, "Third-Generation Technologies for HF Radio Networking," Proceedings of 1998 IEEE Military Communications Conference (MILCOM '98), IEEE*, Boston, MA, October 1998.

[7] SCOPE Command HF E-Mail, Technical Manual - TO 31R2-2GRC244-11

[8] Eric E. Johnson, "Simulation Results for Third-Generation HF Automatic Link Establishment," Proceedings of 1999 IEEE Military Communications Conference (MILCOM '99), IEEE*, Atlantic City, NJ, October 1999.

[9] Eric E. Johnson, "Fast Propagation Predictions for HF Network Simulations," Proceedings of 1997 IEEE Military Communications Conference (MILCOM '97), IEEE*, Monterey, CA.

[10] Eric E. Johnson, "The Walnut Street Model of Ionospheric HF Radio Propagation," NMSU Technical Report.

[11] Zeigler, B.P., "DEVS Today: Recent Advances in Discrete Event - Based Information Technology," supported by NSF Grant No. DMI-0122227, "Discrete Event System Specification
(DEVS) as a Formal Modeling and Simulation Framework for Scaleable Enterprise Design."

[12] Zeigler, B.P., T.G. Kim, H. Praehofer, Theory of Modeling and Simulation, Second Edition, Academic Press. New York, NY, 2000,

[13] Hu, X., Zeigler, B.P., Mittal, S., "Dynamic Configuration in DEVS Component-based modeling and Simulation," accepted and to be published, Simulation: Transactions of The Society for Modeling and Simulation International, November 2003.

[14] Kishor S. Trivedi, Probability and Statistics With Reliability Queuing and Computer Science Applications, Prentice-Hall, Inc., Englewood Cliffs, N.J, 1982.

[15] Reference for Navigation Algorithms - http://www.codeguru.com/Cpp/Cpp/algorithms/general/article.php/c5115/

[16] Andrew S. Tanenbaum, Computer Networks, Third Edition, Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1996.

[17] Tracy Camp, Jeff Boleng, Vanessa Davies, "A Survey of Mobility Models for AdHoc Network Research," Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp. 483-502, 2.

[18] Zeigler, B.P., Mittal, S., "Modeling and Simulation of Ultra Large Networks: A Framework for New Research Directions," supported by NSF Grant ANI-0135530, ULN Workshop, July 2002.

[19] ICEPAC Technical Manual - Ionospheric Communications Enhanced Profile Analysis & Circuit (ICEPAC) Prediction Program Technical Manual

[20] Zeigler, B.P., Multifaceted Modeling and Discrete Event Simulation, Academic Press, London, UK, 1984.