

Access 5 Project Office
NASA
P.O. Box 273
Edwards, CA 93523 USA
661-276-2440
661-276-3880 FAX
www.access5.aero

COVER SHEET

Access 5 Project Deliverable

Deliverable Number: CCC007-Rev2

Filename:

CCC007_C3 System Performance Simulation and User Manual_Rev2_FINAL.doc

Title: C3 System Performance Simulation and User Manual

Abstract: This document is a User's Manual describing the C3 Simulation capabilities. The subject work was designed to simulate the communications involved in the flight of a Remotely Operated Aircraft (ROA)¹ using the Opnet software. Opnet provides a comprehensive development environment supporting the modeling of communication networks and distributed systems. It has tools for model design, simulation, data collection, and data analysis. Opnet models are hierarchical – consisting of a project which contains node models which in turn contain process models. Nodes can be fixed, mobile, or satellite. Links between nodes can be physical or wireless. Communications are packet based. The model is very generic in its current form. Attributes such as frequency and bandwidth can easily be modified to better reflect a specific platform. The model is not fully developed at this stage – there are still more enhancements to be added. Current issues are documented throughout this guide.

Status:

WP – Work in Progress Draft

Limitations on use:

This document is an interim deliverable. It explains the status of the simulation model development as of February 2006. Enhancements scheduled for later in 2006 have not been implemented. Also, the model has not been benchmarked against experimental data.

¹ Please note that ROA is synonymous with Unmanned Aircraft (UA) and Unmanned Aircraft System (UAS).



C3 System Performance Simulation and User Manual

GETTING STARTED
Guidelines For Users

DRAFT Version 2
2/8/2006

CONTENTS

CHAPTER 1 – INTRODUCTION

CHAPTER 2 – PACKET TYPES

CHAPTER 3 – LINE OF SIGHT (LOS) VS BEYOND LINE OF SIGHT (BLOS) COMMUNICATIONS

CHAPTER 4 – REMOTE OPERATED AIRCRAFT (ROA)

CHAPTER 5 – AIR VEHICLE CONTROL STATION (AVCS)

CHAPTER 6 – AIR TRAFFIC CONTROLLER (ATC)

CHAPTER 7 – SATELLITE (GEOSAT)

CHAPTER 8 – SATELLITE GROUND STATION (GS)

CHAPTER 9 – PHYSICAL LINK

CHAPTER 10 – INTRUDER

CHAPTER 11 – COLLECTING STATISTICS AND RUNNING SIMULATIONS

CHAPTER 12 – POTENTIAL FUTURE ENHANCEMENTS

CHAPTER 1 – INTRODUCTION

Opnet Overview

To simulate the communications involved in the flight of a Remotely Operated Aircraft (ROA)² the software Opnet was selected. Opnet provides a comprehensive development environment supporting the modeling of communication networks and distributed systems. It has tools for model design, simulation, data collection, and data analysis. Opnet models are hierarchical – consisting of a project which contains node models which in turn contain process models. Nodes can be fixed, mobile, or satellite. Links between nodes can be physical or wireless. Communications are packet based.

The model is very generic in its current form. Attributes such as frequency and bandwidth can easily be modified to better reflect a specific platform. The model is not fully developed at this stage – there are still more enhancements to be added. Current issues are documented throughout this guide.

Viewing the Model

This manual assumes that the user has access to the software Opnet (version 11 or newer) and has some familiarity with the software. Copy the files that we have provided into your model directory on the computer. Open the Opnet Modeler. Select “File” → “Open”, as shown in Figure 1.

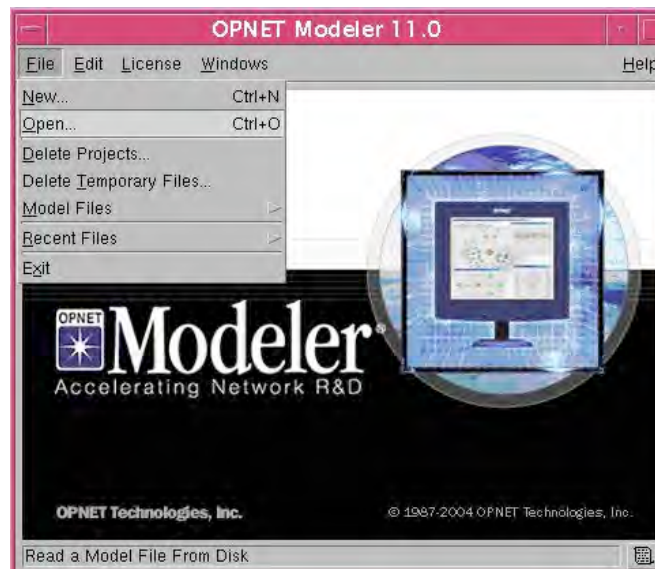


Figure 1 The Opnet Modeler Interface

² Please note that ROA is synonymous with Unmanned Aircraft (UA) and Unmanned Aircraft System (UAS).

C3 System Performance Simulation and User Manual – Getting Started

Select the correct Model Directory on the left and the filename *Access5_Feb06* on the right and then click “Open”. The Project Editor window, similar to that shown in Figure 2, will open. While in the Project Editor, you can view all the nodes and physical links. The *a5* scenario contains seven nodes: Remote Operated Aircraft (ROA), Air Vehicle Control Station (AVCS), two Air Traffic Controllers (ATC, ATC_0), Satellite, Satellite Ground Station (SAT_GS), and Intruder. The scenario contains one physical link between the SAT_GS and the AVCS. Note that the icons for the ROA and Intruder always point West, regardless of their orientation. The *a5_interference_packetgen* scenario will be addressed in CHAPTER 12.

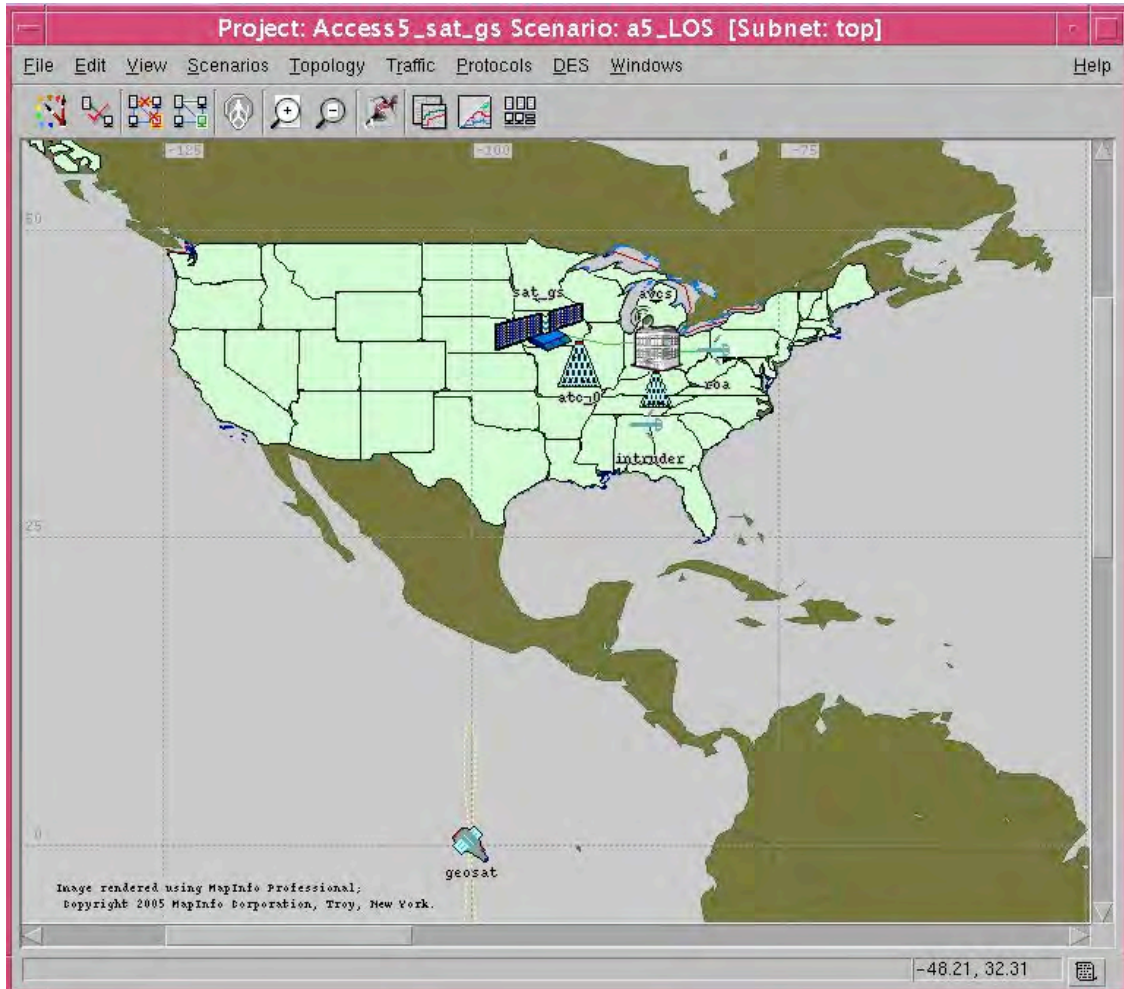


Figure 2 The Project Editor Interface

From the Project Editor you can double click on any of the nodes to view them in further detail in the Node Editor. For example, double clicking on the Geosat node opens the window shown in Figure 3. Right clicking on the antenna, transmitter, or receiver and selecting “Edit Attributes” will open a window with pertinent information. For the antenna the attributes include pattern and target location. For the transmitter and receiver the attributes include channel characteristics, modulation type, and closure model. Double clicking on the other process blocks in the Node Editor will open the Process Editor, as shown in Figure 4.

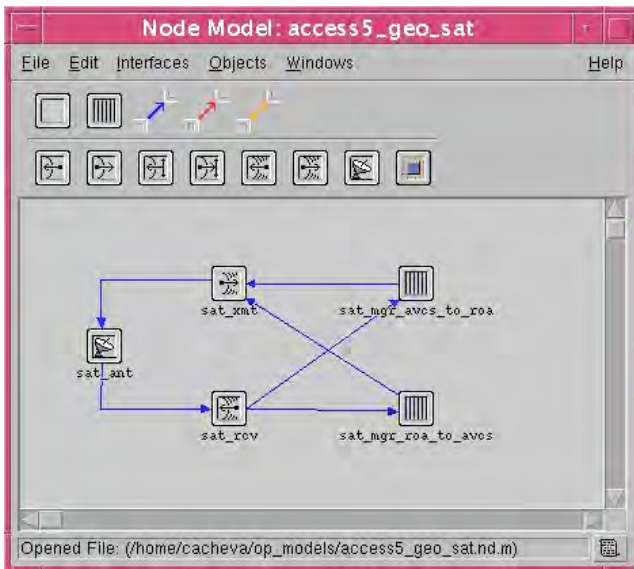


Figure 3 The Node Editor Interface

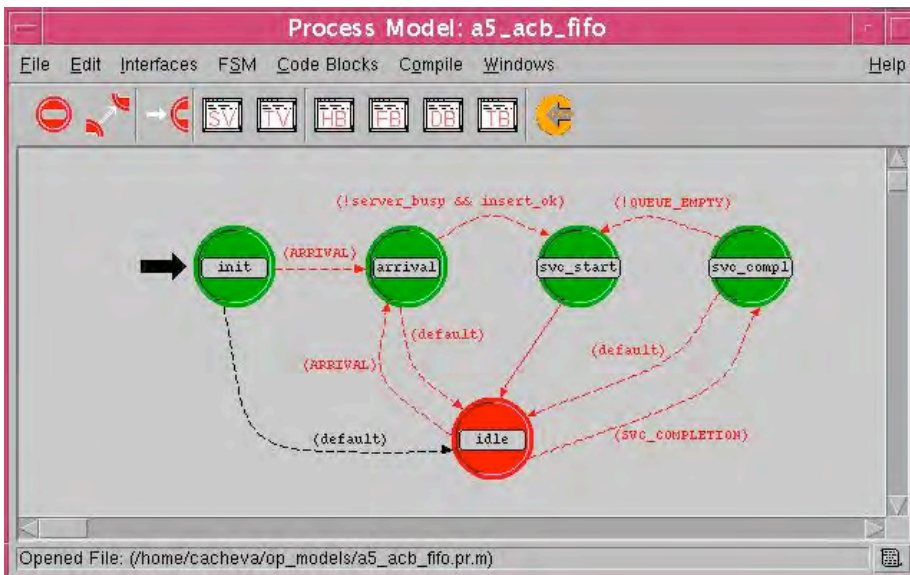


Figure 4 The Process Editor Interface

From the Process Editor, clicking on SV (State Variables), TV (Temporary Variables), or HB (Header Block) allows you to edit the various Process Model variables. The red and green circles in the Process Model are called States. Double clicking on the top half of a state will open a window with the code that is executed when the state is entered. Double clicking on the bottom half of a state will open a window with the code that is executed with the stated is exited. The code for this simulation is written in C.

All of the various editors can be opened from any window. Select “File” → “Open” to open an Open Window. Click on “Files of Type” as shown in Figure 5. Some

C3 System Performance Simulation and User Manual – Getting Started

of the most often used file types are Project, Node Model, Process Model, Link Model, Pipeline Stage (C code), Antenna Pattern, and Packet Format. We have already shown examples of the Project Editor, Node Model Editor, and Process Model Editor. The Link Model Editor is used to edit physical links, such as the one between the satellite ground station and the AVCS. The Pipeline Stage is used to create the code that determines whether communication should be line of sight (LOS) or beyond line of sight (BLOS). The Pipeline Stage also performs a host of other calculations, including propagation delay and bit error rate. The Antenna Pattern Editor allows the user to change antenna gain in the theta and phi planes. The Packet Editor is used to vary the length and content of various packet types, as will be discussed in CHAPTER 2.

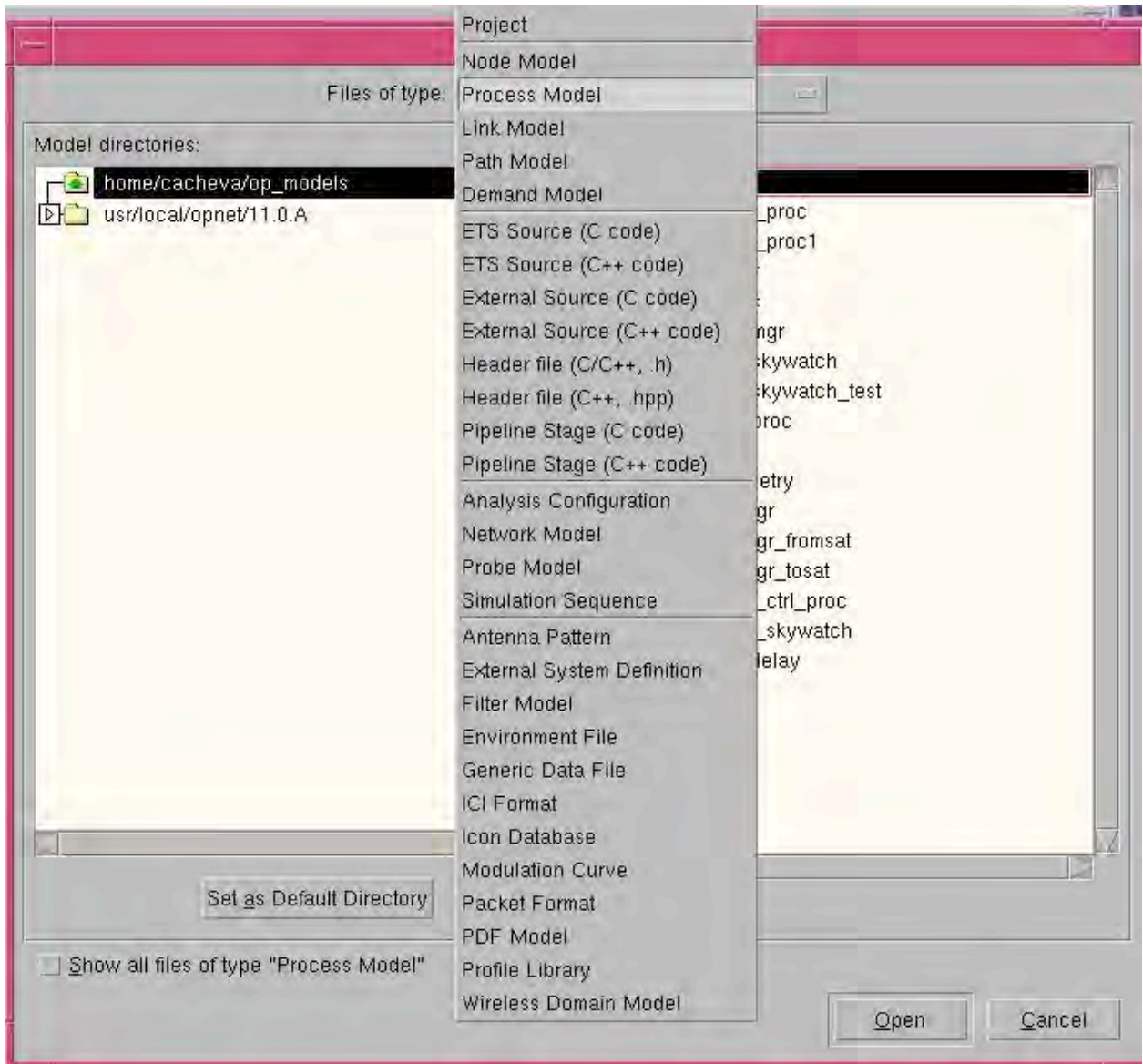


Figure 5 The Open Window showing the type of files that can be opened.

CHAPTER 2 – PACKET TYPES

Opnet communications are packet based. Packets must be created for the various message types. We have created packets for this simulation that are currently serving as placeholders. When more information is available as to how long each message packet is and what content is conveyed the packets can be easily updated. Currently the model has sixteen packet types as shown in the Open window of Figure 6. The Packet Editor is shown in Figure 7 for the *roa_to_avcs_data* packet. As its name implies, this packet type is used to send data from the ROA to the AVCS. Currently the data being sent has been left generic, but as information becomes available the packet sizes can be adjusted and the packet fields can be made more specific to include information like position, fuel remaining, etc. The other packet types are described in Table 1.

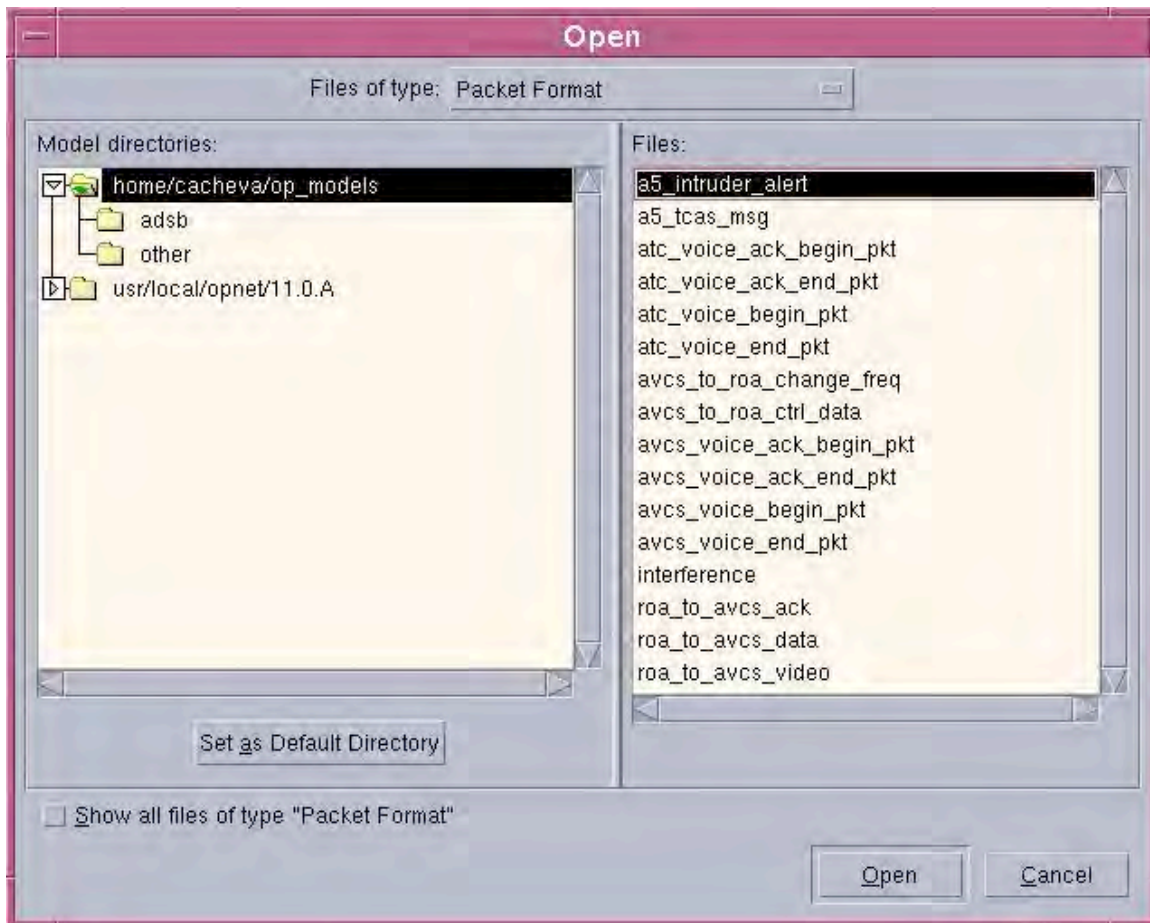


Figure 6 Packet types available in this simulation.

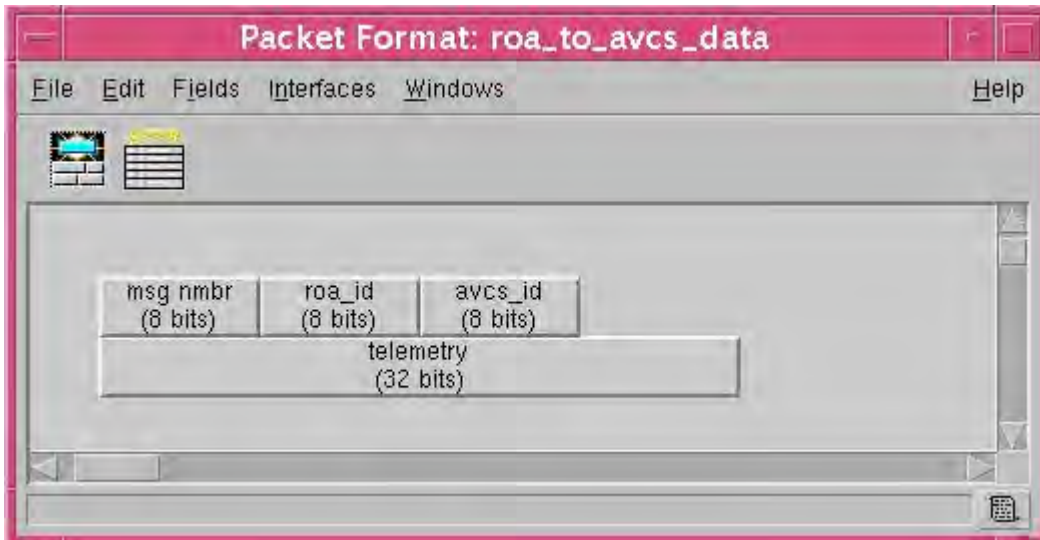


Figure 7 The Packet Editor.

Table 1 Packet Types

<i>Packet Type</i>	<i>Originating Node</i>	<i>Destination Node</i>	<i>Content</i>
A5_intruder_alert	ROA	AVCS	This packet is sent when an Intruder is too close to the ROA
A5_tcas_msg	Intruder	ROA	The Intruder sends the ROA its position
Atc_voice_ack_begin_pkt	ATC	AVCS	The ATC has received voice from the AVCS and begins acknowledging it
Atc_voice_ack_end_pkt	ATC	AVCS	The ATC has received voice from the AVCS and finishes acknowledging it
Atc_voice_begin_pkt	ATC	AVCS	The ATC begins talking to the AVCS
Atc_voice_end_pkt	ATC	AVCS	The ATC finishes talking to the AVCS
Avcs_to_roa_change_freq	ROA	ROA	The ROA sends this packet internally from one process to another when it is changing ATC sectors
Avcs_to_roa_ctrl_data	AVCS	ROA	The AVCS sends the ROA control

C3 System Performance Simulation and User Manual – Getting Started

Avcs_voice_ack_begin_pkt	AVCS	ATC	commands The AVCS has received voice from the ATC and begins acknowledging it
Avcs_voice_ack_end_pkt	AVCS	ATC	The AVCS has received voice from the ATC and finishes acknowledging it
Avcs_voice_begin_pkt	AVCS	ATC	The AVCS begins talking to the ATC
Avcs_voice_end_pkt	AVCS	ATC	The AVCS finishes talking to the ATC
Interference	interference ³	depends on transmitter characteristics	Generate interference in a bandwidth of interest
Roa_to_avcs_ack	ROA	AVCS	The ROA acknowledges received commands
Roa_to_avcs_data	ROA	AVCS	The ROA sends telemetry data to the AVCS
Roa_to_avcs_video	ROA	AVCS	The ROA sends video to the AVCS (this is a placeholder for now)

³ This node is in the *a5_interference_packetgen* scenario to be discussed in CHAPTER 12.

CHAPTER 3 – LINE OF SIGHT (LOS) VS BEYOND LINE OF SIGHT (BLOS) COMMUNICATIONS

Line of Sight (LOS) communications occur when there is closure between the ROA and AVCS. Closure is determined by the antenna patterns of the ROA and AVCS and the distance separating the two nodes. Beyond Line of Sight (BLOS) communications occur when the ROA's antenna is out of range of the AVCS's antenna. Which communication route is required is determined by the Pipeline Stages *a5_closure* and *a5_c2_rxgroup*. *A5_c2_rxgroup* determines which receivers and transmitters make valid pairs. It also prevents a node from receiving its own transmissions, particularly for voice, where the transmitter and receiver use the same frequency. *A5_closure* determines whether there is closure on the LOS route between the ROA and AVCS; if there is not closure it reroutes the packet to the BLOS route. The BLOS route traverses from the AVCS through a physical link to the Satellite Ground Station to the Satellite to the ROA and vice versa.

CHAPTER 4 – REMOTE OPERATED AIRCRAFT (ROA)

The Remote Operated Aircraft (ROA) is a mobile node. To edit its trajectory, right click on the node and select Edit Attributes. The ROA node is shown in Figure 8. It contains a queue named *roa_mgr* which directs the packets within the node. It contains five processes: *roa_collision*, *roa_ctrl*, *roa_atc_freq*, *roa_telemetry*, and *blos*.

Roa_collision receives position data from the Intruder and if the Intruder is too close it forwards a warning to *roa_ctrl*.

Roa_ctrl receives control commands from the AVCS and generates acknowledgements for the AVCS. *Roa_ctrl* also generates alerts for the AVCS when *roa_collision* forwards a warning.

Roa_atc_freq calculates which ATC the ROA is closest to. When the current ATC is no longer the closest, the current ATC sends a voice communication to the AVCS notifying it that the ROA is switching sectors. The AVCS then issues a control command to the ROA to change frequencies. *Roa_ctrl* forwards this command to *roa_atc_freq* which then changes the ROA's receiver and transmitter frequencies to match those of the closest ATC.

Roa_telemetry generates telemetry data for the AVCS on a regular time interval.

Blos sends packets through the *blos* transmitter when closure fails with the AVCS.

The ROA includes an isotropic antenna, receiver, and transmitter for LOS communications named *roa_ant*, *roa_rcv*, and *roa_xmt* respectively. It also includes a directional antenna, receiver, and transmitter for BLOS communications named *roa_blos_ant*, *roa_blos_rcv*, and *roa_blos_xmt* respectively. The ROA has a transmitter solely dedicated to communicating with the ATC, *roa_xmt_atc*. This was necessary because the ROA is assumed to always be LOS with the ATC, therefore its transmitter does not need to use the closure code *a5_closure* that was specially adapted to deal with the possibility of two nodes which can be either LOS or BLOS.

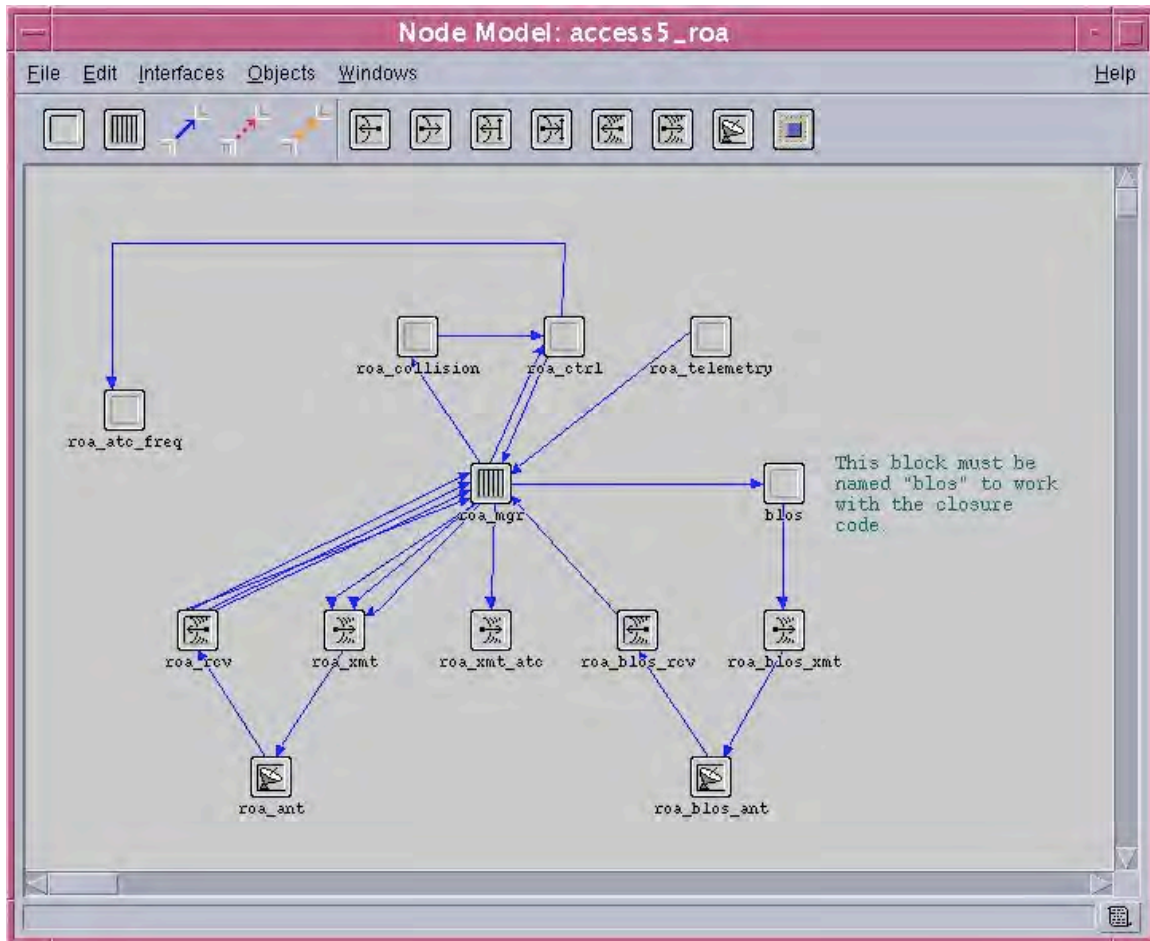


Figure 8 The Remote Operated Aircraft (ROA) Node.

The LOS receiver has four channels. Each receiver channel’s properties (data rate, frequency, bandwidth, modulation type, and packet types) must be identical to the intended transmitter. The first channel is reserved for voice communications from the ATC. The second channel is for incoming control commands from the AVCS. The third channel is for position information from the Intruder. The fourth channel is for voice communications from the AVCS.

There are two LOS transmitters. *Roa_xmt* has three channels. The first channel is reserved for sending intruder alerts and acknowledgements to the AVCS. The second channel is for sending telemetry information to the AVCS. The third channel is for relaying ATC voice communications to the AVCS. The second LOS transmitter, *roa_xmt_atc*, has one channel for relaying AVCS voice communications to the ATC.

The BLOS receiver has one channel and receives all incoming messages when the ROA is BLOS. Its channel properties must match those of the Satellite.

The BLOS transmitter also has one channel for sending all outgoing messages when the ROA is BLOS. Its channel properties must match those of the Satellite.

CHAPTER 5 – AIR VEHICLE CONTROL STATION (AVCS)

The Air Vehicle Control Station (AVCS) is a fixed node. The AVCS node is shown in Figure 9. It contains a queue named *avcs_mgr* which directs the packets within the node. It contains four processes: *avcs_voice*, *avcs_ctrl*, *avcs_telemetry*, and *blos*.

Avcs_voice receives voice transmissions from the ATC and generates an acknowledgement. It also generates voice transmissions for the ATC and waits for an acknowledgement. The time between voice transmissions is based on a Poisson distribution while the length of each transmission is based on a Uniform distribution. To edit the parameters of either distribution go into the Header Block (HB) in the *avcs_voice* process model. MSG_INTERVAL is associated with the Poisson distribution while MSG_DURATION is associated with the Uniform distribution. To change the interval distribution to something other than Poisson, edit the first line of code in the INIT state of the *avcs_voice* process. **Note that at this time the a5 scenario does not contain any interference and the acknowledgments are always received. In the future when interference is implemented, dropped packets will be a concern. At that point it will be necessary to add a timed interrupt such that if an acknowledgement is not received the process repeats the original communication. Otherwise the process will continue listening for an acknowledgement and it will not be able to send or receive voice communications for the remainder of the simulation. Also note that the time delay statistic is not being collected for the acknowledgements.**

Avcs_ctrl sends routine control messages to the ROA and receives acknowledgements from the ROA. When the ROA changes ATC sectors, a packet field is set in the control message which commands the ROA to change the appropriate transmitter and receiver frequencies. *Avcs_ctrl* also receives intruder alerts from the ROA and generates urgent control messages for collision avoidance. Note that there is no collision avoidance logic in this simulation. It is assumed that the predefined trajectories of the ROA and Intruder define any collision avoidance maneuvers that will be necessary during the simulation.

Avcs_telemetry receives telemetry information from the ROA.

Blos sends packets through the *blos* transmitter when closure fails with the ROA.

The AVCS includes an isotropic antenna, receiver, and transmitter for LOS communications named *avcs_ant*, *avcs_rcv*, and *avcs_xmt* respectively. It also includes a point to point receiver and transmitter pair for BLOS communications named *avcs_pt_rcv*, and *avcs_pt_xmt* respectively.

The LOS receiver has three channels. Each receiver channel's properties (data rate, frequency, bandwidth, modulation type, and packet types) must be identical to the intended transmitter. The first channel is reserved for voice communications from the

C3 System Performance Simulation and User Manual – Getting Started

ATC. The second channel is for telemetry information from the ROA. The third channel is for intruder alerts and acknowledgements from the ROA.

The LOS transmitter has two channels. The first channel is for voice communications to the ATC. The second channel is for control commands to the ROA.

The point to point receiver has one channel and receives all incoming messages via the Satellite Ground Station (GS) when the ROA is BLOS. Its channel properties (data rate and packet type) must match those of the GS.

The point to point transmitter also has one channel for sending all outgoing messages via the GS when the ROA is BLOS. Its channel properties must match those of the GS.

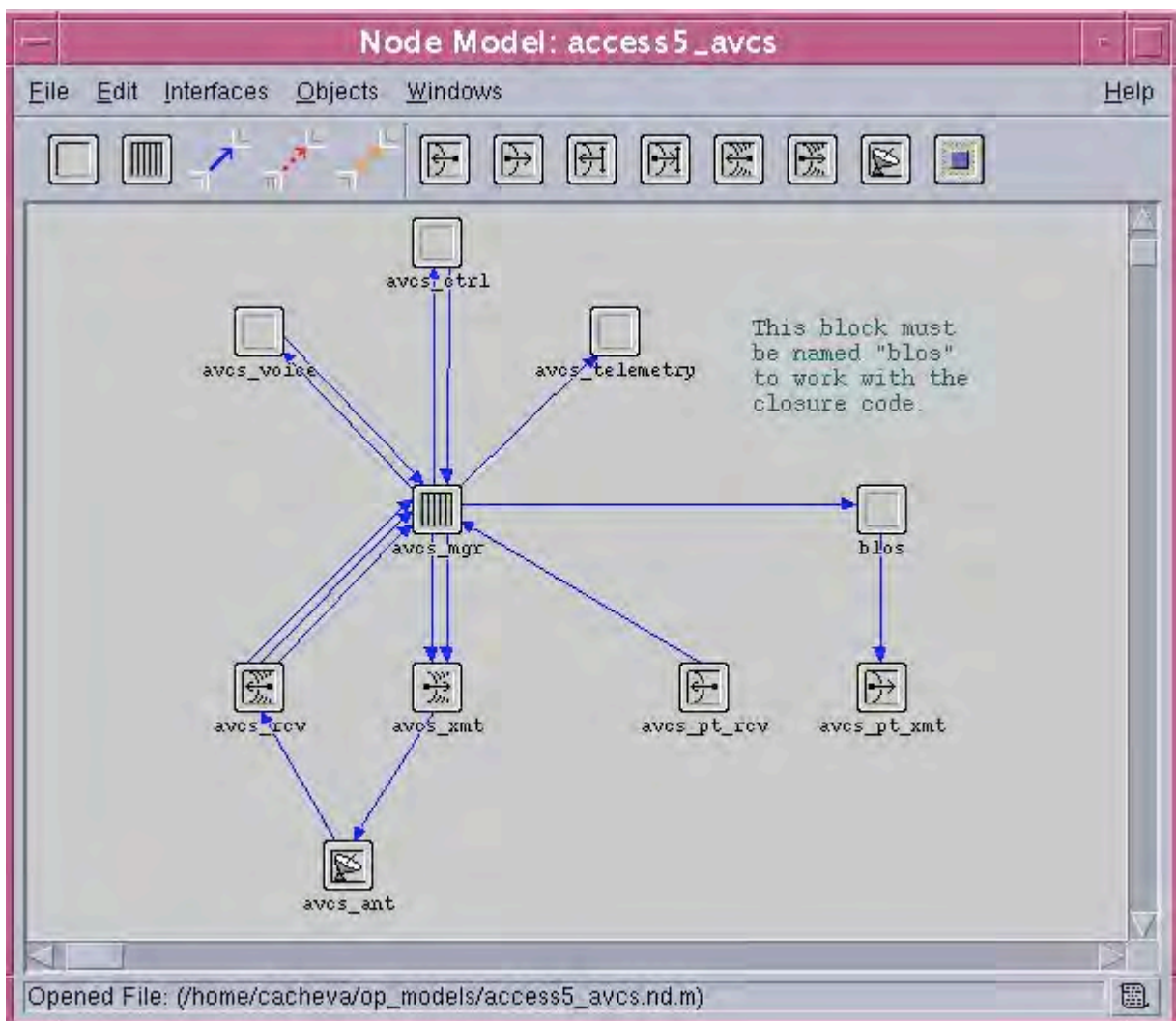


Figure 9 The Air Vehicle Control Station (AVCS) Node

CHAPTER 6 – AIR TRAFFIC CONTROLLER (ATC)

The National Airspace System (NAS) is divided up into sectors. Within each sector is an Air Traffic Controller (ATC). The ATC is responsible for coordinating the movement of air traffic within its boundaries. This simulation model involves a flight that crosses from one sector to another; therefore there are two ATC's (ATC, ATC_0) in the model. Please note that the term 'sector' is used loosely here and does not correlate to the FAA defined sectors, as will be discussed below. Both ATCs are fixed nodes. Both ATC and ATC_0 are identical. Their node model is shown in Figure 10. It contains two processes: *atc_voice* and *handoff*.

Handoff detects when the ROA is leaving its sector and causes *atc_voice* to initiate a voice communication to the AVCS instructing it to change the appropriate transmitter and receiver frequencies.

Atc_voice receives voice communications from the AVCS and generates an acknowledgement. It also generates voice communications for the AVCS and waits for an acknowledgement. The time between voice transmissions is based on a Poisson distribution while the length of each transmission is based on a Uniform distribution. To edit the parameters of either distribution go into the Header Block (HB) in the *atc_voice* process model. MSG_INTERVAL is associated with the Poisson distribution while MSG_DURATION is associated with the Uniform distribution. To change the interval distribution to something other than Poisson, edit the first line of code in the INIT state of the *atc_voice* process. **Note that at this time the a5 scenario does not contain any interference and the acknowledgments are always received. In the future when interference is implemented, dropped packets will be a concern. At that point it will be necessary to add a timed interrupt such that if an acknowledgement is not received the process repeats the original communication. Otherwise the process will continue listening for an acknowledgement and it will not be able to send or receive voice communications for the remainder of the simulation. Also note that the time delay statistic is not being collected for the acknowledgements.**

It is assumed that the ROA is always LOS with the ATC of the sector it is in. Because the ATC is always LOS with the ROA, it only needs one receiver, *rcv*, and one transmitter, *xmt*. Each has one channel for communicating with the ROA. Each channel's properties (data rate, frequency, bandwidth, modulation type, and packet types) must be identical to the intended matching channels in the ROA. There is no antenna defined in this node, so Opnet assumes an isotropic antenna.

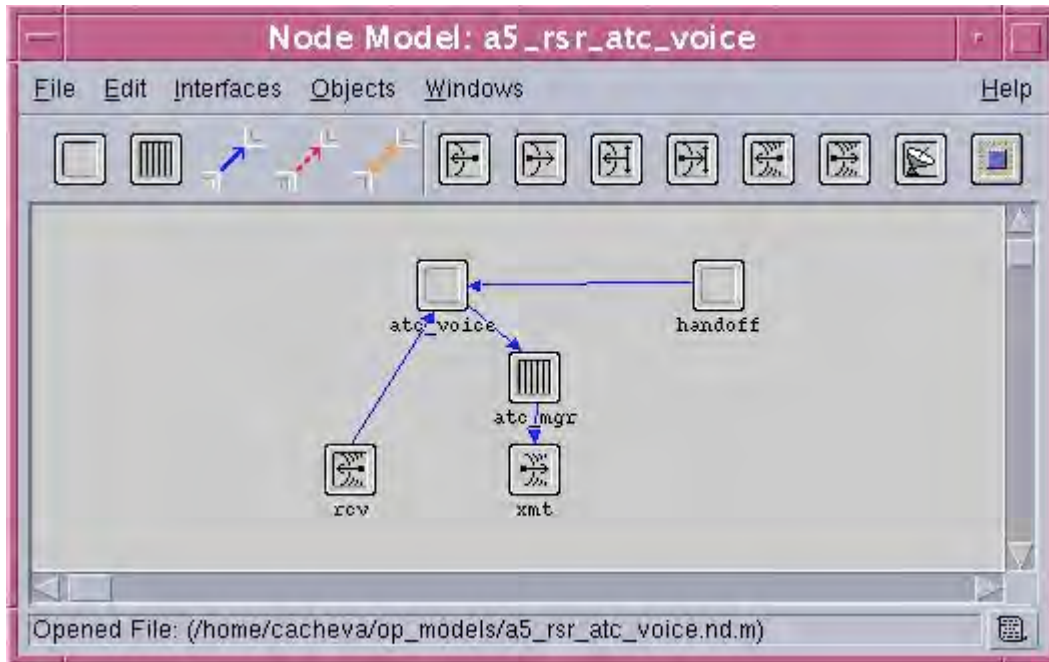


Figure 10 The Air Traffic Controller (ATC) Node

CHAPTER 7 – SATELLITE (GEOSAT)

The Satellite (GEOSAT) node is only used when the ROA and AVCS are BLOS. To state the obvious, GEOSAT is a satellite node. It has a geosynchronous orbit. The GEOSAT node is shown in Figure 11. It contains two queues. The first relays messages from the AVCS to the ROA and is appropriately named *sat_mgr_avcs_to_roa*. The second queue relays messages from the ROA to the AVCS and is named *sat_mgr_roa_to_avcs*. The GEOSAT includes a directional antenna which is pointed at the Satellite Ground Station. Its footprint covers the continental United States. GEOSAT also includes a receiver and transmitter named *sat_rcv*, and *sat_xmt* respectively.

The receiver has two channels. Each receiver channel's properties (data rate, frequency, bandwidth, modulation type, and packet types) must be identical to the intended transmitter. The first channel is for control commands from the AVCS to the ROA and AVCS voice communications which are relayed to the ATC by the ROA. The second channel is for alerts, acknowledgements, and telemetry from the ROA to the AVCS and ATC communications relayed from the ROA to the AVCS.

The transmitter has two channels. The first channel is for control commands from the AVCS to the ROA and AVCS voice communications which are relayed to the ATC by the ROA. The second channel is for alerts, acknowledgements, and telemetry from the ROA to the AVCS and ATC communications relayed from the ROA to the AVCS.

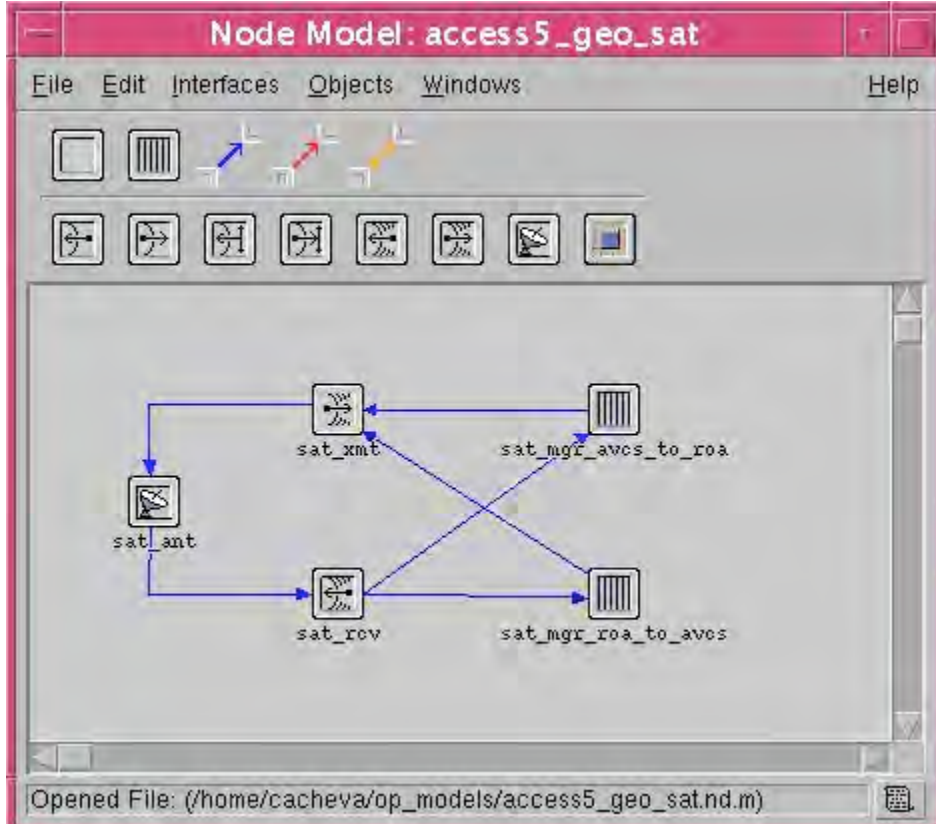


Figure 11 The Satellite (GEOSAT) Node

CHAPTER 8 – SATELLITE GROUND STATION (GS)

The Satellite Ground Station (GS) node is only used when the ROA and AVCS are BLOS. The GS is a fixed node. The GS is shown in Figure 12. It contains two paths.

The first path forwards messages from the AVCS to the GEOSAT. It starts with a point to point receiver *gs_pt_rcv* whose properties (data rate and bandwidth) must match those of the AVCS point to point transmitter. Next it connects to the process *gs_mgr_to_sat* which forwards messages to the transmitter. The transmitter *gs_xmt* has one channel for relaying control commands and AVCS voice communications from the AVCS to the ROA via the GEOSAT. Its properties (data rate, frequency, bandwidth, modulation type, and packet types) must match those of the GEOSAT. The GS also includes an isotropic antenna which both paths use.

The second path forwards messages from the GEOSAT to the AVCS. The receiver *gs_rcv* has one channel for relaying alerts, acknowledgements, telemetry, and ATC voice communications from the ROA via the GEOSAT to the AVCS. Its properties must match those of the GEOSAT. The process *gs_mgr_from_sat* forwards messages to the point to point transmitter *gs_pt_xmt* whose properties must match those of the AVCS point to point receiver.

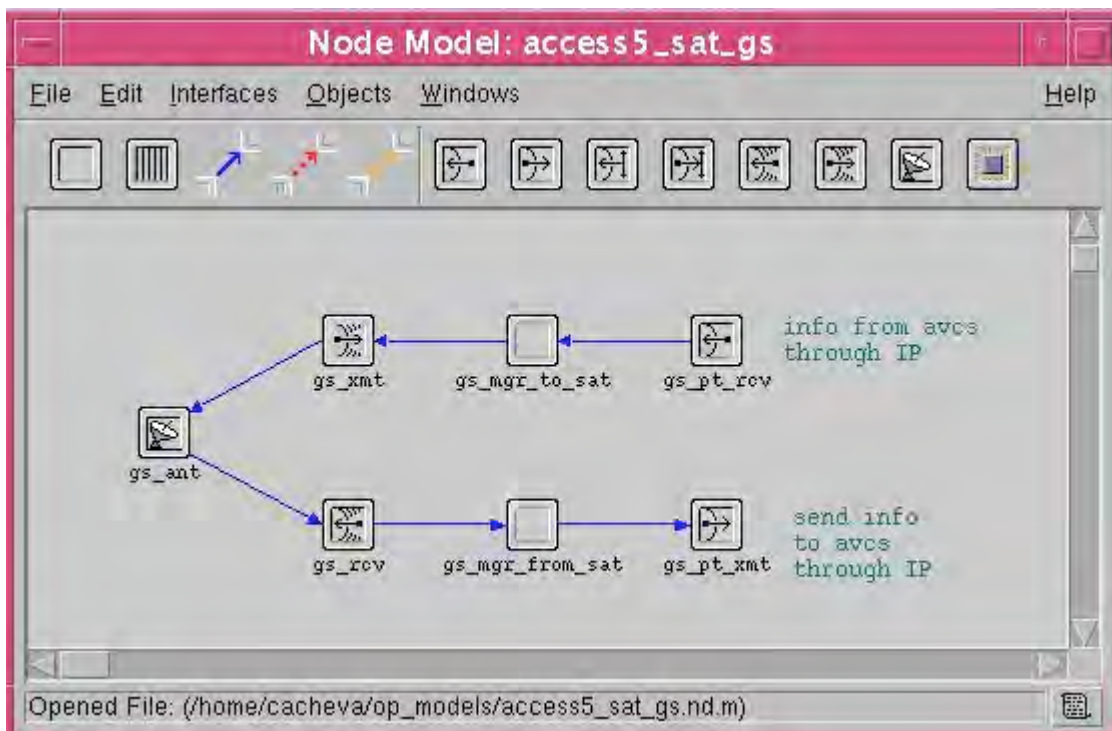


Figure 12 The Satellite Ground Station (GS) Node

CHAPTER 9 – PHYSICAL LINK

The Physical Link connects the AVCS to the GS as shown in Figure 13. It is only used when the ROA and AVCS are BLOS. The Physical Link supports all packet types. The appropriate receivers and transmitters of the AVCS and GS must be defined in the Link's Attributes. In addition to the propagation delay (based on the speed of light) there is also a processing delay associated with the link.

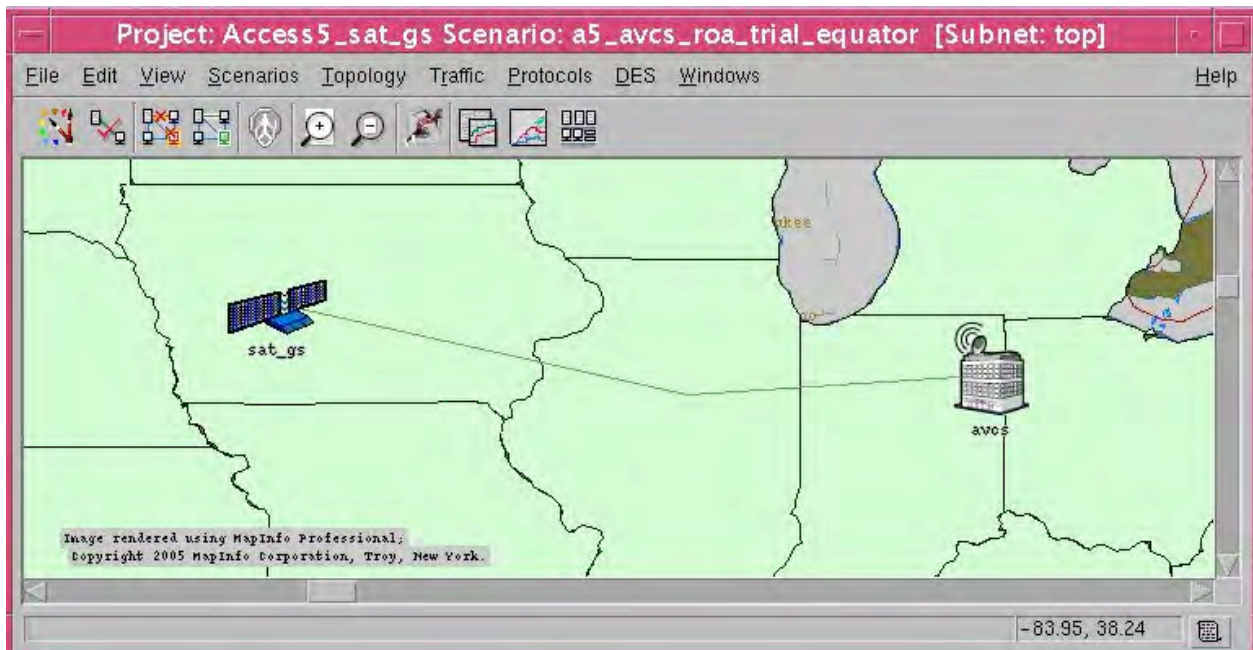


Figure 13 The Physical Link Between the GS and AVCS

CHAPTER 10 – INTRUDER

The Intruder is a mobile node. To edit its trajectory, right click on the node and select Edit Attributes. The Intruder is shown in Figure 14. It contains one process, *intr_collision*, which generates packets with its position data on a regular time interval. The packets are sent to the transmitter *intr_xmt*, and finally to the isotropic antenna, *intr_ant*. The transmitter has one channel whose properties must match those of the LOS receiver in the ROA, *roa_rcv*.

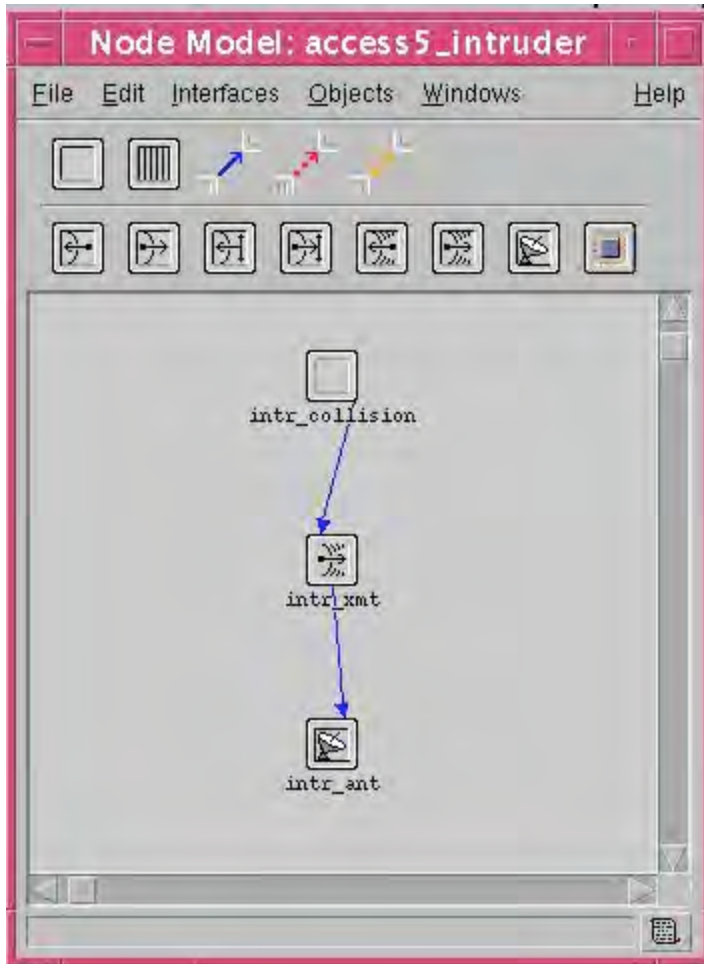


Figure 14 The Intruder Node

CHAPTER 11 – COLLECTING STATISTICS AND RUNNING SIMULATIONS

The following provides instructions on running the simulator and collecting statistics during the simulation. The statistics to collect are specified before running the simulator.

From the Project Editor select “DES” → “Choose Individual Statistics”. Select whichever statistics you are interested in monitoring. Some examples of available Opnet statistics are queuing delay, throughput (packets per second), bit errors, signal to noise ratio, and utilization. **Note that the time delay statistic is not currently available for the voice acknowledgements between the ATC and AVCS.**

Once the statistics are set up you are ready to run the simulation. You must choose the appropriate scenario by selecting “Scenarios” → “Switch to Scenario”. Next, from the Project Editor select “DES” → “Configure/Run Discrete Event Simulation”. Select a simulation input from the tree on the left and fill in all applicable data on the right. Once you have set up the simulation simply press “Run”.

Note that with the current node locations and trajectories a Duration of 40 minutes is necessary to observe the communications switch from LOS to BLOS. Also note that the Intruder approaches and closely passes by the ROA within the first 10 minutes.

After the simulation has successfully run close the Simulation Sequence window. From the Project Editor select “DES” → “Results” → “View Statistics”. Select the results you would like to view from the Discrete Events Graph tree on the left. You can view the selected results in the display on the right.

CHAPTER 12 – POTENTIAL FUTURE ENHANCEMENTS

Should this project be continued in the future, there are a number of features that would significantly enhance this simulation.

Currently the time delay statistic for the voice communications between the ATC and AVCS includes only data collected from initial transmissions. The data still needs to be collected for the acknowledgements.

The queues in all nodes currently process packets received on a first in first out basis. The data should be prioritized such that information regarding safety of flight has the highest priority.

Add a node or nodes that use the same spectrum as the nodes already in the simulation to correspond to interference. Without interference the BER curves will predict better performance than can realistically be expected. Note that this work has been started. A node has been created to generate interference and can be viewed in the scenario *a5_interference_packetgen*.

Once interference has been added, logic for how to handle dropped packets will need to be added. For example, currently voice acknowledgments between the ATC and AVCS are always received. If however the acknowledgment is dropped the node will continue waiting for the acknowledgement and it won't be able to talk or hear any new information for the rest of the simulation. It will be necessary to add a time out function such that if an acknowledgement isn't received in a specified amount of time, the original message is repeated.

Implement procedures for communication security, such as encryption.

Realistically, a pilot can only speak if the channel is free (pilot push to talk event). Right now the AVCS can speak as long as the ATC isn't speaking. It would be more accurate to add a variable delay before the AVCS starts speaking to mimic waiting for the channel to clear.

A very important step in model development is benchmarking. It would be extremely beneficial to compare the model outputs with specific flight tests where the flight parameters and communication data were recorded and are available. By varying the amount of interference and time delays at various stages in the model BER curves and time delay curves can be generated. These results can be compared to the experimental results to determine realistic ranges of interference and the processing delays. These values can be set as the default values in the model. With realistic parameters as the default values, additional parameter can be varied to see how they affect the BER curves

C3 System Performance Simulation and User Manual – Getting Started

and time delay curves. Some parameters of interest are frequency, bandwidth, modulation, transmit power, amount of interference, and packet lengths.

Finally, the model is still “unpolished”. It is the combination of hard work by several people. It would be beneficial to go through all the files and coordinate the naming conventions and comments in the code.