JCSS 7.0

Code of Best Practices

January 2008

Disclaimer: As of October 2007, NETWARS was re-designated by the Program Manager Office as the Joint Communications Simulation System (JCSS). JCSS was selected as the new industry name to better reflect the inherent joint communication capabilities of the software. Users should be aware that no software updates were conducted as part of the software name change.

TABLE OF CONTENTS

NETWARS

1 INTRODUCTION

NETWARS is developed by the Command, Control, Communications, and Computer (C4) Systems Directorate (J-6) of the Joint Staff (JS) in partnership with the Defense Information Systems Agency (DISA) Enterprise Analysis Branch (GE345) to simulate the performance of defense communications systems at the operational level and below. NETWARS provides modeling and simulation (M&S) capabilities for measuring and assessing the information flow through military communications networks (strategic, tactical, and operational). Output from NETWARS provides considerable utility in determining which communications systems might be overloaded during select times within a particular scenario and assists with prudent acquisition planning decisions.

Without the realistic simulations provided by NETWARS, the risks of catastrophic failure on existing and contingency Joint Task Force (JTF), service communications networks supporting operational deployment in the various theaters, and other possible scenarios are largely unknown. NETWARS quantifies these risks and identifies the C4 deficiencies by simulating the effects of various situations and stresses on planned networks in support of warfighter operations. NETWARS is the only modeling and simulation tool under development that will allow the DoD to identify connectivity and throughput problems for JTF-level operations without entering actual mission or battlefield conditions.

NETWARS also provides a toolset that allows users to develop, simulate, and analyze a variety of specific scenarios to answer communication burden issues in the Joint Arena. It can be used to perform—

- Communication burden analysis: the ability to assess the effects of full operational combat traffic loading on tactical information exchange processes and communications networks
- Contingency analysis: the ability to conduct quick-turn communications planning for small regional conflicts and peacekeeping scenarios
- Emerging technology analysis: the ability to evaluate new communications systems and technologies.

1.1 DOCUMENT PURPOSE

The *Code of Best Practices* documents lessons learned and best uses from past studies to provide future NETWARS study teams valid processes, insights, and guidelines. The primary objective is to facilitate future NETWARS-based studies in terms of both study planning and execution.

This guide addresses two NETWARS study team perspectives; the study lead and the hands-on NETWARS analyst. The study lead is defined as the personnel responsible for defining study objectives, developing schedules, assigning resources, staffing personnel, data gathering, and the final product development. The NETWARS analyst is defined as the personnel responsible for scenario development including design, traffic generation, scenario execution, results analysis, implementation and verification. (Note that this guide is intended to describe NETWARS-specific lessons learned rather than general OPNET ITGuru analysis concepts and procedures.)

1.2 WHY USE NETWARS?

There are several reasons why a NETWARS-based study should be conducted. First, the DoD has indicated that simulation will play a significant role in the acquisition of defense-related systems to cut systems acquisition costs, improve reliability, and bring systems into operation more rapidly. Second, there are many areas where simulation can be applied to support software development and acquisition such as requirements specification, process improvement, architectural tradeoff analysis, and product line practices. Third, modeling and simulation technology capable of network analysis needs is now mature, easy to use, of low cost, and readily available.

A NETWARS-based study enables one to address issues before they become insurmountable problems. A NETWARS-based study forces users to think in global terms about network behavior and the fact that networks are more than the sum of their individual components. For example, A NETWARS-based study can provide insights into the designs of processes, architectures, and data traffic before significant time and cost have been invested.

1.3 OVERVIEW OF CONDUCTING A NETWARS-BASED STUDY

The framework for initiating a NETWARS analysis is based on building a scenario. A scenario consists of one or more OPerational FACilities (OPFAC) and their associated communications equipment. OPFACs are the basic building block of a NETWARS scenario and are user-defined groupings of communications equipment that can move together similar to the communications assets of a tank or a plane. OPFACs can also be grouped into organizations to provide military hierarchical relationships and context within the scenario. Traffic can be scripted using an existing text file of IERs, new IERs built in the user interface, imported as flows or application profiles or as a combination of these.

When constructing scenarios, the user interface employs an approach that structures organizations and their associated equipment into a hierarchy that parallels military organizational structure called an Organization. Functional relationships (e.g., command relationships, specific operational support, and peer element) among the various elements are defined by the user. Although users primarily use organizational structures to group the

OPFACs that constitute specific units, they also have the capability to specify the parameters and connectivity of a select set of communication device models (CDM).

Although the methodology within the Scenario Builder is largely concerned with organizational and functional relationships, the actual Simulation Domain deals with the explicit movement of information between the CDMs. A major function of NETWARS is to map these information transfers between each organization's communications devices, simulated traffic sources, and specific communications paths. For more detailed information on how to conduct a NETWARS study, see Sections 2 and 3.

1.4 REFERENCES

In addition to this *Code of Best Practices*, other important NETWARS reference documents include—

- JCSS v7.0 User's Manual, December 2007
- JCSS v7.0 Technical Reference Manual, December 2007
- JCSS v7.0 Software Design Description, December 2007
- JCSS v7.0 System Administrator's Manual, December 2007
- JCSS v7.0 Installation Procedures Manual, December 2007

The NETWARS Program Web site can be accessed at:

- http://www.disa.mil/netwars/

The NETWARS Program maintains support avenues for users with questions or issues associated with the NETWARS software. The NETWARS Help/support can be accessed by:

- Telephone: 240-497-3313 ext. 2699
- E-mail: netwars@opnet.com
- Web: http://www.netwars.disa.mil (See NETWARS Help Information below)

NETWARS Help Information

To access NETWARS Support Center:

1. Go to http://www.netwars.disa.mil
2. Select the *Cancel* button on the logon screen
3. Follow the instruction to complete the password request form
4. A Username/Password will be sent to the email address provided on the request form.
5. Return to the link above, and log on with the Username/Password provided.

The User forum under '*Resources from NETWARS Users'* provides access to posts from previous NETWARS community discussions:

- Click on NETWARS User Forums
- Select a forum
- Click the dropdown button at the top right corner of the screen, and
- Choose topics the desired time frame.

References describing the general use of OPNET ITGuru, NETWARS' Simulation Engine, include—

- OPNET ITGuru 14.0 On-line Documentation.

Other references include—

- *NATO Code of Best Practices for C2 Assessment*, 1998
- *Guidelines for Army Analysts – How To Conduct an Analysis and Present the Results,* February 1989.

NETWARS

2 GUIDE FOR THE NETWARS STUDY LEAD

This section is intended for the lead on a NETWARS-based study. It captures processes and lessons learned in study planning, execution, and management. The NETWARS study lead is responsible for many tasks such as defining study objectives, establishing schedules, defining an overall study approach, developing study-related products, as well as identifying and gathering data required to conduct a study. The following sections provide descriptions of these study lead tasks. In addition to these tasks, the study lead is responsible for study team communications. Often the execution of the study is directly enhanced by effective study team communications. Whereas the perfect study lead would exhibit a robust technical and operational knowledge associated with the study objectives, often the study lead must rely on others to augment his or her skill set.

2.1 DEFINING THE PROBLEM (STUDY PURPOSE)

Before initiating any study effort, the study lead must develop a clear understanding of problems or issues that establish the requirement for a study. The study lead must be able to articulate these problems into a succinct statement known as the problem statement or study purpose. A clearly defined problem statement will identify known issues, personnel affected by these issues, and possible high-level solutions. The study lead should rely on subject-matter experts and other key analysts to clearly develop and refine the problem statement or study purpose.

It is difficult to deny the importance of the problem statement when conducting a study. Too often, problem statements are omitted because they are thought to be obvious. Problem statements contain many implicit assumptions but their benefit is that they allow conflicts to be avoided by helping everyone to focus on the same issues outlined therein.

2.2 GOALS/OBJECTIVES

The key to a timely and meaningful execution of a NETWARS study is the development of study goals/objectives that are derived from the problem statement. It is important to realize that although the study goals/objectives are based on empirical facts, the development of reasonable and flexible study goals/objectives will help the study lead to effectively respond to the study problem. Because a study lead often will limit or hamper the study by establishing rigid objectives and requirements, the following set of questions should be considered:

- Who is the audience or client (e.g., decision makers) for this study and the resulting analysis?

- What previous studies have been completed on similar topics?
- What impact will possible study conclusions have on the audience?
- What will the final study report or briefing look like?
- What are the minimum and maximum expectations of the study audience?
- What time and financial constraints might have an impact on this effort?
- What are the technical limitations of NETWARS (i.e., size of scenario and total amount of traffic)?

Once the study lead answers the questions, focus can be placed on defining manageable tasks in order to accomplish the stated goals and objectives. These tasks should be detailed and decomposed into the specific steps needed to accomplish stated goals and objectives.

2.3 SCHEDULE AND COST

Scheduling is an essential early task in any type of NETWARS study. There are many reasons to develop a schedule for a NETWARS study. First and foremost, scheduling is a mechanism in which the progress of the study is measured. More often than not, the schedule should be the primary concern of project management. A timely delivery may be as important as functionality or quality in determining the ultimate value of a study. The situation can be complicated by the fact that the delivery date may have been determined by external constraints rather than by the inherent size and complexity of the study. The result can be an overly ambitious schedule. Given that schedule is such a key concern, it is critical for the study lead to monitor adherence to intermediate milestone dates; early schedule slips are often a precursor to future problems. It is also critical to have objective and timely measures of progress that provide an accurate indication of current status and that can be used for projecting the dates of future milestones.

A reliable schedule possesses the following characteristics—

1. A historical database of events
2. Structured processes for estimating component size and reusability
3. A mechanism for extrapolating from demonstrated accomplishments from past studies
4. Audits trails
5. Data collection and feedback processes that foster capturing and correctly interpreting data from work performed
6. Structured processes for design/implementation changes (imposed changes are acceptable only when legitimate design-to-cost or schedule-to-cost processes are followed).

Project duration is one of the vital parameters used to construct new cost models or to calibrate existing ones. The NETWARS study lead must understand what the duration includes and excludes. If a project took 3½ months, it would be reasonable to ask exactly what was included in that time period: Did it include system requirements analysis and design or just the NETWARS activities? Did it include model integration and testing or just the model integration?

Tracking milestone dates and deliverables provides a macro-level view of a project schedule. As noted earlier, slips in the early reviews and deliverables are often precursors of future problems. A much greater visibility can be achieved by tracking the progress of activities, which culminate in reviews and deliverables. By tracking the rate at which the underlying units of work are completed, one gets an objective basis for knowing where the project is at any given point in time and a basis for projecting where it will be in the future.

2.4 RESOURCES AND STAFFING

The NETWARS study lead is tasked with identifying the types and sources of information required for a completion of all study goals/objectives within the study schedule. For each study data requirement, the study lead should develop a unique data-gathering approach. The data-gathering approach or plan should identify a primary and a secondary data source. Because the most difficult data sources often are external personnel, the study lead should determine the appropriate organizations or agencies that could provide information and engage critical personnel as soon as possible.

2.4.1 Roles and Responsibilities

A NETWARS study requires a wide range of personnel skills. An effective study team is paramount to the success of any NETWARS-based study. It makes little difference how encompassing the problem statement and the goals/objectives are if the study team is poorly staffed. The staffing requirements necessary for each study are directly related to the complexity of the study. At a minimum, the study lead will have to understand the functional roles necessary to complete a study and identify an individual(s) for each role. In the early stages of the study, the mix of study team personnel should be about evenly divided between the data-gathering/design team and the implementation/analysis teams. As the study transitions from the development of study goals/objectives to data gathering to data analysis, the data-gathering/design team members must be replaced almost one-to-one with telecom analysts to build up the analyst team. It is recommended that at least one full-time task or project lead be on board for every four (4) team members to keep all of the tasks coordinated.

2.4.1.1 Study Lead

The study lead should have a broad understanding of communication principles and of what NETWARS is and how its capabilities can be used. The study lead should also understand the perspective of the study audience and develop a level of operational or technical expertise that helps them to better understand the study problem. Most importantly, the study lead must understand and must be able to clearly articulate the link between the problem statement, the goals/objectives, the study approach, and the study results in order to recruit the staff needed to execute each step of the study.

2.4.1.2 Data Gathering

During a NETWARS study, the data-gathering team uses research to lay the groundwork for obtaining the goals and objectives of the study. The study team must first identify credible sources of data and then develop an approach to collect the necessary data to satisfy the study's goals and objectives. The data must then be presented to the design team in a format that is easily translated into a system design. Further discussion of the duties of data gathering can be found in Section 2.5.

2.4.1.3 Scenario Design and Implementation

The implementation team must have had some previous experience with NETWARS or OPNET products such as ITGuru. The implementation team must have experience in developing scenarios, manipulating model attributes, executing simulations, and verifying the integrity of simulation results. Sections 3.1 and 3.2 discuss the process of implementing the system/network design within NETWARS.

2.4.1.4 Simulation Analysis

The analysis team is responsible for collecting, assimilating, and presenting the final results of a study. This entails understanding the scenario, data inputs, and the associated simulation results. The analysis team must develop an approach to present the results that satisfy the study goals and objectives. The analysis team should have a basic understanding of statistical analysis and experience with conducting some type of analysis. See Section 3.6 for a detailed discussion of analyzing NETWARS results and developing an analysis approach.

2.4.2 Hardware Specifications

Beyond the suggested administrative needs of running a NETWARS-based study, adequate hardware is essential to using NETWARS. Machines with a minimum of a Pentium IV, 1GHz processor, 1GB of random access memory (RAM) and 20 GB of hard disk drive (HDD) should be available for actual scenario execution. Although this exceeds the actual specifications needed to successfully develop OPFACs and scenarios in NETWARS, these specifications are required for the actual runs. The processor speed is needed to keep reasonable run times. The RAM is needed to handle the amount of packets that will flow through a typical network system. A hard drive being used as RAM is not a good substitute because it adds considerable run time and tends to make the software unstable. The hard drive space is necessary because any normal-sized network will produce over 1 GB worth of data per run, and most studies will require multiple runs. Spreadsheet and database programs are needed to store designs, maintain configuration control on scenarios, and provide the ability to parse and manipulate the large amount of resulting data.

2.5 DATA GATHERING

Data gathering should support the level of detail and/or abstraction required to meet the different study objectives and should verify the ability of the NETWARS system to support study requirements through existing libraries and to identify new development tasks. The primary role of the study lead during data gathering is to collect initial information that provides a foundation for the study design, NETWARS implementation, and analysis approach. This first round of data includes—

- **Operational Plans (OPLANs)** – These plans will help the study team develop an operational tempo for the study scenarios. The operational tempo will help scope all other study data requirements.
- **Network Topology Diagrams** – These diagrams should define the different study equipment and the connectivity of the equipment. This information is critical to a timely development of a study scenario. Network topology diagrams can be presented via PowerPoint presentations and/or Microsoft Visio products. The study lead should work with the study analyst to ensure that all required information is identified and gathered.
- **Traffic data** – Traffic data should be categorized as either critical IERs or background traffic. Background traffic can be extracted from network management systems and network monitoring devices. The study lead should identify sources of network traffic data and work with the study analyst to develop an approach to use this data in NETWARS. At a minimum, the study lead should understand the network context under which the data was collected and the format used to store the data.

This information provides a framing of the organization and its network that allows the study team to identify—

- The appropriate range of questions to be used in conducting further data probes
- Personnel within the organization to be interviewed
- Supporting documents to be reviewed.

Once the data gathering process is completed, the study team can finally put the project into context with respect to the relationship between the Problem Statement, Goals/Objectives, Schedule and Cost, and Resources and Staffing. The study lead should be prepared to adjust study goals/objectives based on the availability of supporting data. Any changes to the goals/objectives should be discussed with the study team. To avoid any misconceptions that could impact the credibility of the study results, the study lead should be able to clearly explain the impact to the study audience.

2.6 STUDY APPROACH

The study lead should understand the relationship between the study goals/objectives, the design and implementation of the NETWARS scenario, and the analysis approach. The study analyst will handle most of the technical issues associated with the execution of the study, but the study lead should understand the high-level approach, to ensure that the study goals/objectives are achieved. Figure 1 (next page) shows a high-level relationship between the data-gathering products, the NETWARS implementation, and the simulation and analysis.

To begin, the study lead should convene a kickoff meeting. The purpose of the kickoff meeting is to determine the study schedule, define the study objectives, and identify the team members. Section 2.3, *SCHEDULE AND COST*, and Section 2.4, *RESOURCES AND STAFFING,* provide insight into these issues.

When identifying the study requirements the study lead should consider what the main study questions are. For example, is the study to answer bandwidth utilization questions (a Capacity Planner question) or to determine the impact of a new application on an existing network (a Scenario Builder question)? Questions such as this should be answered as soon as possible because they will drive the design of the architecture in NETWARS. See Section 2.1, *DEFINING THE PROBLEM (STUDY PURPOSE),* for more details.

The next step is to define the scenario by gathering all the required data, Section 2.4.1.2, *Data Gathering*, provides some insight into this process. Also, there may be an OPLAN or some other like documents that will help to define the study requirements in terms of organizations, OPFACS, and their interconnections.

Once the Study Lead has a good understanding of the requirements, development of the architecture and traffic can begin. Section 3.1, *System/Network Design*, and Section 3.3, *Traffic*, provide insight into network and traffic development. The study lead should consider conducting an In Progress Review (IPR) before any real development starts. The IPR helps to ensure that all the team members are aware of the study progress and any remaining issues, such as new model development, before proceeding.

One of the biggest issues a Study Lead may have is when the study requires the development of new communications device models (CDM). The Study Lead must insure that the schedule allows for the development and testing of any new CDMs before they are integrated into the NETWARS architecture. Lastly, the implementation of the architecture should be modular. Develop and test the architecture in sections before combining them. This approach will keep implementation issues localized so the entire architecture need not be examined every time there is an implementation issue. See Section 3.2, *IMPLEMENTATION*, and Section 3.4, *SCENARIO VERIFICATION*, for more details.

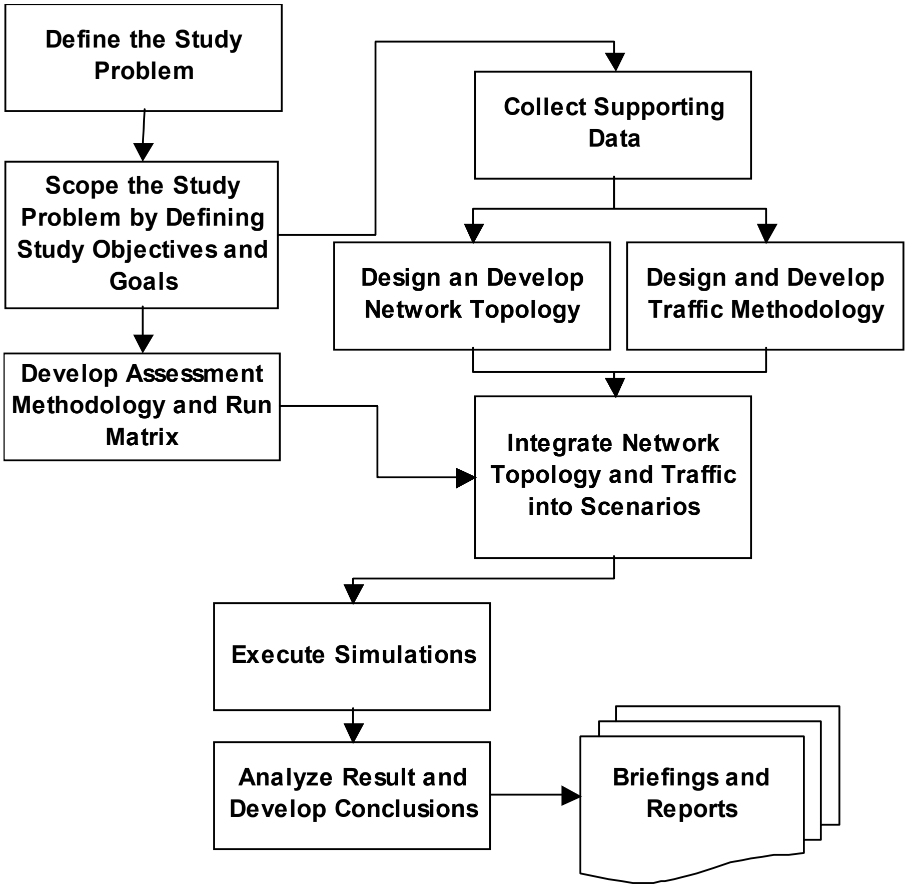

Define the Study Problem
Scope the Study Problem by Defining Study Objectives and Goals
Develop Assessment Methodology and Run Matrix
Collect Supporting Data
Design an Develop Network Topology
Design and Develop Traffic Methodology
Integrate Network Topology and Traffic into Scenarios
Execute Simulations
Analyze Result and Develop Conclusions
Briefings and Reports

Figure 1 Study Approach

When developing traffic, the Study Lead needs to understand the different types of traffic and the issues associated with each type of traffic. See Section 3.3, *Traffic*, for details.

Once the initial architecture and traffic have been developed, the study lead should consider conducting another IPR. The purpose of this IPR is to ensure the Study Lead has an understanding of and agrees with implementation of both the architecture and traffic before the simulations begin.

Lastly, the simulations and analysis methods used will depend on the study goals; see Section 3.6, *ANALYZING NETWARS RESULTS.* For example, if the study question is to determine the worst-case bandwidth utilization the Study Lead may need to only determine the peak traffic hour (using Microsoft Excel or some other application) and run the Capacity Planer, using the traffic for that hour. However, because discrete event simulations require much more time, the Study Lead can choose to use the Capacity Planner results to focus the analysis by running the simulations only during the peak hours and focusing on the over-utilized links.

2.7 STUDY PRODUCTS

The study lead should develop a set of study products that help the study audience understand the purpose of the study, the study data, the study approach, and the results or conclusions. The study lead should use a combination of reports, briefings, and NETWARS files to communicate with the study audience.

2.7.1 Reports

A NETWARS study report should contain the following basic sections:

- Executive Summary – Summarizes the entire study effort and clearly states the study findings with respect to the overall study purpose.
- Introduction – Provides the project background and a high level view of the assessment approach. It also provides an introduction to the remainder of the study report.
- Study Problem and Objectives – Clearly defines the study problem or purpose and the study goals/objectives. The study lead should provide background to help the study audience understand the operational context of the problem. The study lead should also discuss key study assumptions and provide a brief overview of the entire study approach.
- Topology Development – Describes the high-level design and final NETWARS implementation. The implementation section should describe relevant components such as OPFACs, organizations, links, and IERs.
- Traffic Development – Describes the processes used to gather and develop NETWARS traffic.
- Assessment Methodology and Scenarios – Describes the project scenarios, simulation runs, and subsequent analysis. The scenario descriptions include the methodology of using the topology and traffic to develop the scenarios.
- Analysis and Conclusions – Describes the study findings and provides conclusions relevant to the study questions. The study lead should include statistical significance for each table or graph. This will enhance the credibility of the study results.

2.7.2 Briefings

The study lead should conduct briefings with the study audience to ensure that the study approach is consistent with the audience's expectations. At a minimum, the study lead should prepare the following three key briefings:

- **Kickoff Briefing**: The kickoff briefing starts off the study effort. Study objectives, schedule, and participants should also be discussed.

- **In-Progress Report (IPR)**: An IPR provides program management insight into how the study is progressing. The length of the study will usually dictate how many IPRs are needed.
- **Final out Briefing**: The final out briefing presents the results of the study to the leadership. Classification of results should be considered in advance so that proper precautions can be followed.

2.7.3 NETWARS Files

The study lead should work with the study implementation team and the analyst to identify key NETWARS scenario components and simulation results that should be presented or delivered to the study audience. Future NETWARS studies or additional study-related analyses would be greatly enhanced by a well-structured and organized set of NETWARS files. At a minimum, the following NETWARS files should be organized and distributed for any study:

- Projects and scenarios
- OPFACs
- Organizations
- IER text files
- Simulation results
- CDMs
- Analysis tables and graphs.

The study lead can learn more about these files in the analyst section (Section 3).

NETWARS

3 GUIDE FOR THE NETWARS USER/ANALYST

The NETWARS user or analyst is faced with many challenges in designing, implementing, and analyzing a scenario. This section is designed to help the NETWARS analyst understand and overcome many of these challenges. It focuses on lessons learned in designing, implementing, verifying, executing, and analyzing a NETWARS scenario.

The NETWARS analysts should work closely with other study members to understand all study objectives and associated data. If possible, the analyst should provide technical recommendations that would assist the study lead in developing study objectives and determining the applicability of study data.

The NETWARS analyst should be proficient with NETWARS software and should possess a basic understanding of statistical analysis. Previous experience with OPNET or other discrete event analysis efforts provides a basic foundation for individuals without direct NETWARS experience.

3.1 SYSTEM/NETWORK DESIGN

Analysis requirements are the basis for solving the problem statement, but the design phase is critical to satisfying the requirements. This phase involves a combination of flexible thinking, creative reasoning, and educated guesswork. Many analysts, eager to build models to solve problems, often leave out this critical part of the study process. The design phase will allow the study team to reason through the entire problem and will help prevent the problem from being forced into the context of a particular methodology.

System/network design has two phases. The first, preliminary design, deals with the decomposition of the system/network into large, integrated objects. The products of the data-gathering phase conducted by the study lead and other study team members should drive the preliminary design. The analyst should work with team members responsible for data gathering to fill in holes or gaps in the preliminary design. The second phase is called critical design. In this phase, attributes and methods are more complex and are specified at the level of individual OPFACs, organizations, links, IERs, MOPs, and so forth. This is also where a study can realize the reuse of objects (i.e., OPFACs, system elements, links) because it is possible to guide the design so that lower level objects correspond exactly to those in existing NETWARS libraries, or to develop objects with reuse potential. The design phase should also contain a contingency plan that allows the study team to revisit the system design in the event of unforeseen problems.

The analyst should validate the critical design with the study lead and possibly the study audience. To facilitate the validation of the design, the analyst may decide to document or visualize the design with tools such as Microsoft Visio and PowerPoint or to develop a high-level NETWARS representation.

3.2 IMPLEMENTATION

The implementation phase is the process of identifying model components and integrating them on the basis of network topology diagrams and system/network designs. Using the insight gained from previous phases, analysts identify and create OPFACs and organizations—the building blocks used to create scenarios.

Analysts should develop and maintain study-specific projects for consolidating all work associated with a given study. Projects are basically containers for all project-specific files, including scenarios and simulation results. NETWARS scenarios are network architectures that can represent a specific time phase or course of action. For the remainder of the document, the term scenario will be used to represent scenarios, time phases, and courses of action. The following sections describe the key NETWARS scenario components.

3.2.1 OPFACs and Organizations

The fundamental building block of NETWARS is the OPFAC, a collection of communications device models. OPFACs typically represent a set of devices co-located and assigned to a military unit (e.g., platoon). They are created and modified using the Scenario Builder. Organizations are aggregations, or hierarchies, of military units, and represent the infrastructure of command/support relationships between units and their communications elements, including broadcast networks and point-to-point links. In NETWARS, organizations are a combination of OPFACs and are created and modified using the Scenario Builder.

The analyst should establish for OPFACs and organizations a naming convention that is consistent with study objectives and easily understood by individuals outside the study. Scenario component names should be minimized to avoid excessive output filenames that may confuse future users. Users may often choose to establish the naming convention during the planning of the study scenario. The most difficult naming convention is usually associated with OPFAC development. OPFACs rely on the following three separate character strings for identification within the Scenario Builder.

- **Name –** This string differentiates one OPFAC from another and cannot contain spaces.
- **Functional Name –** This string and the description are used by the Scenario Builder to help the user identify a specific OPFAC within the treeview. The system does not prevent OPFACs from sharing identical functional names and descriptions, so the analyst should prevent confusion by defining unique names for different OPFACs. In addition to its role in the Scenario Builder, the functional name is used by the IER database or text file importation. The IER producer or consumer fields are compared with the functional name instead of the OPFAC name, to allow users to generate multiple IER instances

across numerous OPFACs. Relationships are required between the producer and consumer OPFACs in order import IERs from either a text file or the IER database.

- **Description –** This string and the functional name are used by the Scenario Builder to help the user identify a specific OPFAC within the OPFAC library portion of the treeview.

Before the inclusion of CDMs within an OPFAC, the analyst should examine the different object palettes to identify the appropriate device representation. The *JCSS 7.0 Communication Model V&V Report* can help the user understand the NETWARS standard military communications model, while OPNET ITGuru documentation can help the user understand OPNET commercial off-the-shelf (COTS) models. The analyst should remember that a NETWARS CDM is developed to represent some portion of real life system or device functionality. The fidelity and intended use for a CDM helps the study analysts understand the degree to which the CDM captures or represents the real life functionality of the system. Although no simulation or model is perfect, the analyst tasked with system acquisition or system design should understand the degree of simulation fidelity and explain to the study audience and client all assumptions and limitations.

When developing OPFACs, the analyst should understand the number and type of different CDM interfaces to avoid non-functioning architectures. In addition to model interfaces, each model contains a set of model attributes. The *NETWARS Model Development Guide* establishes requirements for models, including a set of common attributes such as equipment type and classification. Many attributes, however, directly affect model behavior. These attributes should not be adjusted unless the analyst is intimately aware of the impact.

A CDM is developed to represent some portion of real-life system or device functionality. The fidelity of the CDM helps the study analysts understand the degree to which the CDM captures or represents the real-life functionality of the system. Although no simulation or model is perfect, the analyst tasked with system acquisition or system design should understand the degree of simulation fidelity and explain to the study audience and client all assumptions and limitations.

OPFACs often contain multiple CDMs connected with intra-OPFAC links. These links connect devices within an OPFAC. For example, a 10BaseT between a router and a computer both of which are contained within an OPFAC is an intra-OPFAC link. The analyst should always verify the consistency of these links before saving the OPFAC.

The NETWARS analyst often will group the OPFACs into two distinct categories: producer/consumer and intermediate. The producer/consumer OPFACs contain end-system devices such as workstations or phones that allow the analyst to associate the production or consumption of IERs. To ensure intended IER generation and consumption associated with an OPFAC, the analyst should compare the classification of the end-system devices and the classification of associated IERs.

Depending upon the complexity and redundancy of the study, the analyst should define template organizations using a set of existing OPFACs. Template organizations allow the analyst to quickly replicate identical architecture components and to include organizations into different scenarios. For example, a template organization with the Secure Internet Protocol Routing Network (SIPRNET) components of the Bahrain Standardized Tactical Entry Point (STEP) site could be used by different study efforts. Template organizations promote re-use of study components and improve the usability of the NETWARS system for future analysts.

3.2.2 Infrastructure

NETWARS scenarios contain four types of infrastructure (i.e., relationships, point-to-point links, satellite links, and broadcast networks). The NETWARS analyst can specify all four of these types through the Define Infrastructures portion of the Scenario Builder scenario treeview. The following are simple definitions for the four different infrastructure types:

- **Broadcast Networks** – Are networks of radio system devices that are tuned to the same frequency and belong to same frequency hop group.
- **Satellite Links** – Defines a satellite “channel” between two OPFACs through one OPFAC that contains a satellite device model.
- **Wired links** – Represent actual physical media between two nodes
- **Radio links** – Defines the connection between a radio transmitter–receiver channel pair
- **Relationships** – Associates unit relationships between OPFACs

3.2.2.1 Relationships

The primary purpose of relationships is to associate relevant traffic loading to a scenario from either the IER database or IER text files. Therefore, the analyst must define the appropriate relationship between each producer/consumer pair identified for a study. This process can be extremely tedious and requires that the analyst completely understand the relationship requirements of all intended study IERs. For example, if the associated IER text file contains fifty (50) different IERs, each with a unique relationship between two OPFACs, the analyst should verify the existence of each required relationship within the scenario to guarantee the flow of IERs from producer to consumer. The analyst should take care not to define duplicate relationships between OPFACs, because the NETWARS system will associate duplicate IERs by default. For example, defining relationships through a "mesh"-like methodology will generate numerous unintended relationships that will trigger duplicate IERs. The NETWARS analyst should instead define relationships in a one-by-one manner based on the requirements of the final IERs.

3.2.2.2 Point-to-Point Links

Point-to-point links can be either terrestrial or wireless. Therefore, the analyst should understand the devices contained in the two associated OPFACs, to ensure valid configurations. Specifically, users should manually assign the device ports for any link with the following devices: Promina, Satellite Terminals, or Circuit switches. These devices allow the user to “map” specific circuits or paths that require the user to define each port configuration.

3.2.2.3 NETWARS Tactical Radios

3.2.2.3.1 Radio Broadcast Links

Radio broadcast networks are associated with tactical radio devices such as the Single Channel Ground & Airborne Radio (SINCGARS), Falcon, and HaveQuick. The analyst is required to plan frequency allocation and hop groups for relevant systems to avoid unintended interference.

The analyst should also use the line-of-sight range utility in the Scenario Builder to avoid nonfunctioning configurations. Radios that are out of range from each other will not properly transmit and receive information. No error message will be reported in a simulation, so it is very important the analyst understands to place radios within LOS or closer to ensure proper radio communication.

3.2.2.3.2 Sending Data from NETWARS Tactical Radios

All NETWARS Tactical Radios (excluding jtids) are placed in a similar equipment string to send data traffic. The radio must be connected to an Internet Controller (INC) device. The INC allows for data from the IP network to be sent over the radio network. The SINCGARS_INC and Harris_6010 models are both INC devices that can be connected to a radio and also connected to a workstation or server (via point-to-point connection) to send data. A broadcast network must also be configured between the two radios.

3.2.2.3.3 EPLRS Simple Radio Configuration

is EPLRS radios in NETWARS require additional configuration to be used in a scenario. The analyst must deploy a correct equipment string such as provided in section 3.2.2.3.2 of this document. The analyst must create a broadcast network between the EPLRS radios. The broadcast network must be of type "eplrs_broadcast_X" (where X is a number between 0 and 7).

Since EPLRS radios can reside on up to eight different channels (or eplrs_broadcast networks) or use a point-to-point connection between radios, a row in the "INC Interface Mapping Table" must be present for each connection. This table can be found in the attributes of the EPLRS radio. Each row of this table must indicate what port on the radio the Internet Controller (INC) is connected to, and either which broadcast network to use or which point to point port to use. The broadcast network number will match the broadcast network type that was set when creating the broadcast network itself. If the radio has been configured to be used in point to point mode, this value can be left at N/A and the port to be used for the point to point connection should be specified in the third column. Please not that JCSS 7.0 introduces a new high fidelity EPLRS model (EPLRS RS). Further information on this model can be found on the EPLRS Model User Guide.

3.2.2.3.4 Trajectories with NETWARS Tactical Radios

A simulation license is required for NETWARS to take into account the trajectory of a unit in a simulation. Simulated movement of units with NETWARS tactical radios that are part of a broadcast network behaves in one of two ways. If the units are within range of one another, they will be allowed to send information to each other for the duration of the simulation, regardless of their distance from other units. If the units are not within range of one another, they will not be allowed to send information to each other for the duration of the simulation, regardless of their distance from other units. This is an important note to take for an analyst who wishes to study the effects of movement on tactical radio based networks.

3.2.2.4 Satellite Links

To model satellite systems, NETWARS relies on earth terminals and satellite space segments. These devices allow users to develop satellite links or "channels." Satellite links require two OPFACs with satellite terminals and one OPFAC with a satellite device. To help ensure a realistic Satellite Command (SATCOM) configuration, analysts should develop a plan similar to the Satellite Access Request (SAR) that defines the home satellite and uplink transponder for each terminal, the associated bandwidth, and the terrestrial interface for each satellite link. The analyst is required to select the terrestrial port and the uplink for each satellite terminal. This requires detailed planning to ensure that the study scenario accurately portrays the SATCOM architecture. To assist in the planning process, the analyst must understand that each satellite terminal is associated with only one uplink and define an approach that satisfies each pair. Table 1 presents a sample SATCOM planning chart that contains the key components the analyst must define for a study-wide SATCOM architecture.

OPFAC A	Device	Terrestrial Port	Uplink Transponder	OPFAC B	Device	Terrestrial Port	Uplink Transponder	Satellite	Size (K)
MacDill	GSC-39	0	0	BAHRAIN	GSC-52_EA	0	1	DSCS EA	768
MacDill	GSC-39	1	0	JFACC	MSC-74EA	0	2	DSCS EA	512
JFACC	TSC-85IO	0	1	BAHRAIN	GSC-52_IO	0	0	DSCS IO	512
JFACC	TSC-85IO	1	1	JFLCC	TSC-85IO	0	2	DSCS IO	512
BAHRAIN	GSC-52IO	1	0	JFSOCC	TSC-85	0	3	DSCS IO	256
BAHRAIN	GSC-52IO	2	0	CVBG	SHF_Terminal	0	4	DSCS IO	128
ESKAN	TSC-85EA	0	3	Belvoir	GSC-52_EA	0	4	DSCS EA	512
ESKAN	TSC-85IO	0	5	JFLCC	TSC-85IO	1	2	DSCS IO	512
ESKAN	TSC-85EA	1	3	MacDill	GSC-52_EA	2	0	DSCS EA	512
JFACC	MSC-74EA	1	1	Northwest	GSC-52_EA	0	5	DSCS EA	1536
JFLCC	TSC-85EA	0	3	Belvoir	GSC-52_EA	1	4	DSCS EA	1544
BAHRAIN	GSC-52EA	1	1	Belvoir	GSC-52_EA	2	4	DSCS EA	1544

Table 1 – Sample SATCOM Architecture Planning Chart

Satellite terminals marked with “wTSSP” are enabled with Tactical Satellite Signal Processor technology. TSSP can be considered as a synchronous Time Division Multiplexer-Demultiplexer that is part of the satellite terminal. Its function is to multiplex digital input signals into one composite data stream for transmission and to demultiplex one or more received composite data streams into individual data streams.

This is very important when one earth terminal has connections to multiple networks. A ‘circuit’ must be established between each port of the source earth terminal and the destination earth terminal. This can be done simply by right-clicking the satellite link and selecting “Create TSSP Circuit”. The analyst will then specify the source and destination ports. NETWARS will display the device connected to each of these ports in brackets next to the port name.

Analysts may also notice that there are limited bandwidth options when it comes to deploying a TSSP satellite link. The bandwidths that are available are the only bandwidths that can be used by the corresponding earth terminals. If a non-TSSP bandwidth is required, the analyst should consider using non-TSSP earth terminals to create the satellite link.

3.2.3 Collaborative Planning

In some cases the study lead will not have all the details required to complete the architecture in NETWARS. In these cases, the study lead can assign subordinate planners to provide those details using the collaborative planning process. When using the collaborative planner the lead planner must still have a good enough understanding of the architecture to implement it generically in NETWARS. When assigning owners, it's recommended the subordinate planners be given edit privileges to organizations instead of only OPFACs. This gives them the ability to work freely inside the organization. If not, they will only be allowed to modify the assigned OPFACs.

3.2.4 Routing Protocols

NETWARS supports all OPNET standard routing protocols such as Open-Shortest-Path-First (OSPF), Border Gateway Patrol (BGP), and Enhanced Interior Gateway Routing Protocol (EIGRP). The default protocol is Router Interior Protocol (RIP) therefore planners who intend on using the DES function need to be aware of the 16-hop limitation of RIP.

Information regarding protocol implementation can be found in the OPNET ITGuru documentation. To help the analyst implement a realistic approach, this section provides a brief overview of these protocols. All routing protocols currently fall into one of three classes: Distance-Vector, Link-State, or Hybrid. Each class uses its own characteristics and methods to determine the best route to a destination. Distance-Vector and Link-State protocols are best suited for interior routing, whereas Hybrid protocols address the needs of border routes.

3.2.4.1 Distance-Vector Protocols

Distance-Vector protocols are the simplest of the lot. In their most basic forms, each router maintains a database of the different networks to which it can route, and each route in the table has a metric associated with it. The router determines the best path using this metric, which essentially indicates the distance from one router to any other reachable router on the network. In the case of Routing Information Protocol (RIP), these metrics are the number of routers a packet has to go through to reach the destination network. A *hop* occurs each time a packet goes through a router. The route with the least number of hops to the destination network is determined to be the best route. RIP has a maximum allowable hop count of fifteen (15) by default, meaning that sixteen (16) is deemed unreachable. RIP works well with small networks but is inefficient on large networks with slow wide area network (WAN) links or on networks a large number of routers installed.

Another Distance-Vector protocol is the Internet Gateway Routing Protocol (IGRP). IGRP is a Cisco-proprietary distance-vector protocol. This means that all the routers in the network must be Cisco routers to use IGRP. IGRP also uses complex formulas to determine its routing metrics. IGRP has a maximum hop count of 255 with a default of 100, thus overcoming the inherent problem of using RIP with larger networks. IGRP also uses bandwidth and delay of the line by default as a metric for determining the best route to the destination network. This is known as a composite metric. Reliability, load, and maximum transmission unit (MTU) can also be used, although they are not used by default.

The first Distance-Vector protocols had problems detecting erroneous routing information, which can occur during link failures. In some instances a router would be left with information in its routing table that would not be deleted until the router exceeded its maximum hop count (also known as counting to infinity). Counting to infinity may seem as if it would take forever, but for a router using RIP, it is the amount of time before a route's metric is set to a number greater than 16, indicating that the route is unavailable. In the worst-case scenario, it could take as long as 5 minutes for the network to converge.

To deal with these problems, "split horizons" were added to Distance-Vector protocols. Split horizons describe the concept that there is no reason to share routing information about one’s route to a particular host with a router between oneself and that host. For example, a router in New York would not share with a router in Washington, D.C., information about its routes to a router in Miami. To further speed convergence, "triggered updates" occur the instant the network topology changes, rather than at the next scheduled routing-table update.

In general, Distance-Vector protocols do not offer provisions for detecting multiple router loops involving more than one hop. During these types of situations, routers are left counting to infinity before they can readjust their routes. The exception to this is EIGRP, which employs an algorithm to detect loops.

3.2.4.2 Link-State Protocols

Link-State protocols answer the failings of Distance-Vector protocols. This type of protocol maintains a copy of the network map and the links involved. Instead of relying on each router to make the decision about the best route to a destination, Link-State protocol routers share this map with each other to ensure that all routers are in agreement about the best routes between networks.

When changes occur in the topology, the router that detects the change floods the changed information to its neighbor routers, and these routers flood the information in turn to their neighbors. This allows each router to adjust its database and to quickly converge. To allow this flooding to work, each update is time stamped to ensure that a newer update is used if, for some reason, an older update arrives after a newer one.

To recover from failed and restored links, version numbers are attached to each link in the database. Because it would be inefficient for routers to exchange their entire databases after a link comes back as a failure, when the adjacent routes are being brought back up, the routers share version-number information about links. After doing so, they request from the adjacent router a full copy of "interesting" records—records for which the version number of the link is greater than their own.

Because all Link-State protocols maintain a database of the entire network, they are not prone to loops. This is ensured by the use of the Dijkstra’s algorithm to calculate the shortest paths between networks. An example of a Link-State protocol is OSPF, which features multipath routing, load balancing, and least-cost routing and is the successor to Distance-Vector protocols on the Internet.

3.2.4.3 Hybrid Protocols

Hybrid protocols use aspects of both Distance-Vector and Link-State protocols. For example, EIGRP is a hybrid routing protocol that uses additional advanced features to avoid loops and speed convergence. EIGRP uses a method in which each router not only calculates the best current route to a destination network but also determines alternative routes that can be used if the current route fails.

3.2.5 Topologies Based on RCIs

NETWARS provides a module based on the OPNET Technologies Multi-Vendor Import (MVI) feature (now referred to as XDI) to import router configuration information (RCI) files from Cisco Works and/or Juniper. RCI files contain detailed information that allows the user to accurately recreate complex topologies while ensuring detailed network information such as IP addresses and routing protocol characteristics are accurately defined. The process can greatly increase the efficiency of an analyst who wants to build a scenario based on an existing network configuration. More information regarding the details of the process can be found in the NETWARS Users Manual and in OPNET Technologies MVI Use Case documentation.

3.3 TRAFFIC

NETWARS relies on a variety of different mechanisms to simulate the impact of traffic across a given architecture. The analyst must understand the requirements of the study and the possible sources for traffic information.

3.3.1 IERs

The NETWARS IER contains seventeen (17) attributes that define unique characteristics. A common approach that an analyst may choose is to view IERs as two distinct types: critical and background. Critical IERs are directly related to the study objectives and represent mission and task applications that are the focus of the study effort. Background IERs are the additional traffic loading associated with the scenario architecture but are not directly related to the critical IERs.

The development of critical IERs depends on identifying and cultivating valid data sources. The analyst can rely on military doctrine, subject matter experts, and military personnel interviews for sources of critical IERs. Each data source has its advantages and disadvantages. The analyst should work with the study lead to identify the impact of assumptions associated with critical IER development.

IERs contain seventeen (17) fields or attributes—three (3) required and fourteen (14) optional—which define the characteristics of the IER. Tables 2 and 3 list the required and optional attributes and the attribute description.

Table 2 – Required IER Attributes

Attribute	Description
Traffic Type	NETWARS supports three types of traffic: data, voice and VTC.
Size	Defines the number of seconds for voice and VTC IERs or bytes for data IERs.
Average Interarrival	Defines the mean associated with a distribution to determine a generation rate of the IER.

Table 3 – Optional IER Attributes

Attribute	Description
IER ID	String of characters that uniquely defines each IER. The analyst should develop a unique naming convention for each critical IER associated with a study.
Classification	Classification setting that allows NETWARS to select an appropriate producing and consuming device that is consistent with the security requirements of the IER.
Perishability	The limit of time that the IER should be received within. This value is only used through post processing of the IER results and does not affect simulation performance.
Priority	The priority of the IER that reflects the urgency of the IER.
Equipment	Defines the generic equipment type for production of the IER. OPFACs can contain many different pieces of equipment; this value allows the analyst to specify a type of device for IER production.
Consuming OPFAC Name	Defines the functional name of the consuming OPFACs.
Distribution Type	Defines the type of distribution used with the distribution mean to determine a generation rate. NETWARS supports constant, exponential, and uniform distribution.
Start Time	Defines the beginning of a window with respect to simulation time during which IERs will be generated.
Stop Time	Defines the end of a window with respect to simulation time during which IERs will be generated.

Attribute	Description
Producer Device	Defines the device that generates the IER within the producing OPFAC.
Consumer Device	Defines the device that consumes the IER within the consuming OPFAC.
Transport Protocol	Defines if the IER will be sent using TCP, UDP, or N/A.
Unit Relationship Code (URC)	Two characters that define a military relationship between the producing and consuming OPFACs.
Message	Allows the analyst to describe the IER. The analyst should use the description to share additional information such as IER purpose with other study members and future analysts.

The generation of IERs is a series of rules and simple logic. The NETWARS system uses the IER attributes in an attempt to produce a set of traffic exchanges. The IER distribution type and mean provide the key components of the production process. NETWARS uses these values and the random seed generator to establish a set of generation times. The generation window defined by the start and stop times can limit the set of IER generation events. If a specific generation event is triggered by the simulation and is within the generation window, the system uses the equipment type and the classification to choose a device for producing that IER event within the OPFAC. Each device within an OPFAC has a specific generic equipment type and classification that is used by the simulation to select appropriate devices. The producing device then generates a traffic event with the exact size and precedence. The perishability is only used as a post-processing element to help the analyst determine a timely delivery of the event. Perishability has no effect on the generation or transmission of an IER event.

In addition to the automated IER generation process, the analyst should develop a plan to predict or control the devices selected with an OPFAC. This plan will help the analyst ensure that specific devices in the study architecture are selected for each IER event.

3.3.1.1 Threaded IERs

To properly represent the condition events associated with most network applications, the analyst should develop threaded IERs. Threaded IERs are a sequence of critical IERs with condition relationships. For example, OPFAC A sends a critical IER, IER #1, to OPFAC B. If and only if OPFAC B receives IER #1 from OPFAC C, then OPFAC B will send a critical IER, IER #2, back to OPFAC A. Threaded IERs allow the analyst to more accurately reflect tasks and missions through complex information exchanges. Therefore, the analyst should develop threads for those critical IERs that reflect key tasks and missions associated with the Study Objectives and Problem definition.

The analyst should also be careful not to trigger two critical IERs when they use threads. This can occur if the user develops a thread with an existing critical IER and does not set the IER start time attribute to the flag value of THREAD. To avoid this potential problem, the analyst should clearly identify each element or critical IER associated with a thread and set the IER start time to this special flag value.

3.3.1.2 Methodology

The NETWARS IER methodology does not allow the analyst to establish conditional relationships or threading between IER events within NETWARS. Because many critical applications and missions rely on numerous information exchanges, the methodology prevents the analyst from studying complex information exchange sequences. The analyst can attempt to represent these complex chains of events by scripting IER generation with constant distributions and small generation windows. Although this methodology does not allow the analyst to study the specific event sequence, it does allow the user to decompose the performance of each segment of the sequence.

Once the analyst has examined the traffic date, a set of IERs that represents the impact of that traffic must be developed. The analyst might want to develop the minimum number of IERs required to represent the desired traffic characteristics, which will in turn minimize the resulting processing burden. This process of minimizing the number of IERs while maintaining the integrity of the data can be very subjective and should be coordinated with the study lead. The replication of real-life data can be quantified by examining the mean, variance, peak, and goodness of fit characterization. The following is a simple methodology that can be customized to fit different study expectations.

To replicate the mean utilization for a period of time between two devices, the analyst can define an IER based on the following formula.

IER size = Data Bandwidth x Target Utilization ***x*** *Inter-arrival Rate*

The inter-arrival mean should match the given variance of the real-world data. High real-world variance results in higher inter-arrival means. This correlation has been verified via running multiple scenarios using NETWARS. The analyst can use this formula to generate an IER for each time period within a scenario. For example, the analyst could generate twenty-four (24) IERs to replicate the given utilization between two devices for each hour. The analyst should use IER start and stop times to facilitate the scripting of the background traffic from one time period to the next.

3.3.2 Application Profiles

In addition to IERs, traffic can also be represented as applications (e.g., VTC, email, and FTP) using the “Application Config” utility inside the configuration OPFAC. Once the applications are defined, their behavior during the simulation must be set using the “Profile Config” utility inside the configuration OPFAC. After the applications have been configured they can be deployed inside the NETWARS architecture using the Application Deployment Editor inside the Protocols menu. More information regarding the details of the process can be found in the NETWARS Users Manual and in OPNET Technologies documentation.

Files collected using software probes (e.g., Ethereal) can also be imported as applications into NETWARS. However, they must first be filtered and formatted using OPNET's Application Characterization Environment (ACE). ACE provides users the ability to filter captured files so they contain only the application(s) of interest. It also provides a custom interface where users can examine transactional diagrams of an application and the relationship between the users. Once the applications are filtered in ACE, they can be imported directly into NETWARS. However, for this to be an effective methodology the study lead must know the host IP address of the applications of interest. More information regarding the details of the process can be found in Appendix D.

3.3.2.1 Advanced Application Profiles

Defining applications on your network requires two utility nodes to be present in the configuration OPFAC. These two nodes are the "Application Config" and the "Profile Config" utility. The analyst must define each application in the Application Definitions table found in the Application Config utility to best describe how each application behaves. Changing the values in the application definitions table will change how the application behaves and how much traffic is sent. Further information regarding setting up application and profile configuration utilities can be found in Appendix D.

After defining each application the profiles configuration utility must then be defined. A profile can be thought of as a user. Each profile or user can utilize one or more applications (defined in the applications definitions utility). The analyst should carefully select which applications will be contained within the profile. Once all the applications have been selected there are several advanced attributes for the profile that can be modified in order to better represent the user. They are: profile operation mode, start time, duration, inter-repetition time, and application repeatability. Each of these attributes is critical to the behavior of the profile. See below for a definition of each of these attributes and how they will ultimately effect your simulation.

Profile Operation Mode - Defines how applications will start.

- Serial (Ordered) - Profiles can start one after each other in an ordered manner (first row to last row). Choose this method if you plan on only allowing this profile to be run once at a time.

- Serial (Random) - Profiles can start one after each other in a random manner. Choose this method if you plan on only allowing this profile to be run once at a time in random time intervals.

- Simultaneous- Profiles can start all at the same time. Choose this method if you plan on allowing this profile to be two or more times at the same instant in time. This would be useful if you'd like to represent more than one user on a node at a certain time.

Profile Start Time – The profile start time defines when during the simulation the profile session will start. For example, a typical user will not produce much or any traffic until 8AM or the beginning of the work day. If you are simulating a 24 hour day, you may want to set the start times of your profiles to when the user actually sits at his/her machine to produce network traffic. If no traffic is sent between 0000 and 0800 it is suggested to assume 0800 will be the beginning of your simulation.

Profile Duration

- Defines the maximum amount of time allowed for the profile before it ends. Remember, the profile is a container for which the applications will run in. If the application inside the profile has an application duration that is longer than what is set in the profile duration, the application will get cut short.
- When set to 'End of Simulation' profile is allowed to continue indefinitely until the simulation ends.
- When set to 'End of Last Application' profile is allowed to continue till the last instance of an application running as part of this profile ends. If the application repeatability is unlimited, the profile will end when the simulation ends.
- Profiles that repeat should not have their duration set to 'End of Simulation'. If this is done, the application will never repeat.

Profile Repeatability - Specifies the parameters used to repeat execution of profile

- When set to 'Once at Start Time' profile will only be executed once until the duration of profile is met
- When set to unlimited, profile will repeat infinitely
- When attribute is edited, user is allowed to specify
  - Inter-Repetition Time
  - Number of Repetitions
  - Repetition Pattern

Profile Repeatability – Inter-Repetition Time

- Inter-Repetition Time
  - Defines when the next session of the profile will start.
  - Serial - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session completed.
  - Concurrent - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session started. When set to concurrent, the mean outcome should not be zero. A mean of zero would case sessions to be created at an infinite rate.

Now that we have discussed how it is possible to change the repeatability of each profile, we can now discuss how to change the repeatability of each application within each profile. Remember, the profile is a container for which the applications will run in. Make sure to set up the profile correctly so all applications within the profile can run successfully. Often times, users will run multiple applications at different times, with random repeatability. In order to set up how each user or profile runs each application, open the applications table from within the profile configuration table. You will see that each application has 3 major attributes that can be changed in order to effect how they are run. See below for a list of these attributes and their definitions to understand how they will effect your simulation.

Application Start Time Offset

- This attribute has two interpretations based on the value specified for the "Operation Mode".

- If the 'Operation Mode" is set to "Simultaneous", this offset refers to the offset of the first instance of each application (defined in the profile), from the start of the profile.
- If the "Operation Mode" is set to "Serial (Ordered)" or "Serial (Random)", this offset refers to the time from the start of the profile to the start of the first application. It also serves as the inter-application time between the end of one application to the start of the next. If an application does not end (e.g., duration set to 'End of Profile'), subsequent applications won't start.

Application Duration

- The maximum amount of time allowed for an application session before it stops running. The simulation engine will generate traffic defined by the application flow (based on OPNET application algorithms) for the entire duration of the application.
    - When set to 'End of Profile', the application will end when the profile duration has expired.
    - When set to 'End of Last Task', the application will end when the last task of the application has completed regardless of task completion times.

Application Repeatability

- Similar attributes to profile repeatability
- Specifies the parameters used to repeat applications within the surrounding profile.
    - Inter-repetition Time
        - Defines when the next session of the application will start depending on the Repetition pattern.
    - Number of Repetitions
        - Specifies the distribution name and arguments to be used for generating random session counts.
    - Repetition Patterns
        - Defines when the next session of the application will start.

Application Repeatability – Inter-repetition Time

- Serial - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session completed.
- Concurrent - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session started. When set to concurrent,

the mean outcome should not be zero. A mean of zero would case sessions to be created at an infinite rate.

Application Repeatability – Repetition Pattern

- Serial - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session completed.
- Concurrent - The start time of the next application is computed by adding the inter-repetition time to the time at which the previous session started. When set to concurrent, the mean outcome should not be zero. A mean of zero would case sessions to be created at an infinite rate.

3.3.3 Background Traffic

The analyst can use empirical traffic collected from probes and network management systems to define background traffic. Both of these sources provide much of the same information, but each source also has unique fields that add rigor to the background loading processes. Hardware probes can provide detailed packet information such as IP addresses and protocol information, whereas network management systems can provide link utilizations. The level of fidelity of the data from both sources is dependent upon the capture characteristics. The normal rule is that the more frequent the data capture, the more accurate the background traffic representation. There are however, some lessons learned when importing traffic that is important to remember:

1. In order to use probe data, an IP mapping table of the entire network is required. For NETWARS to sort the probe data properly, every IP address must first be assigned as an alias.
2. The Study Lead must have an understanding of all local area networks (LAN) of interest. For example, how the LAN connects to the point of presence (POP) router.
3. When using the MVI process to import traffic there will always be IP addresses NETWARS will be unable to map. This is caused by traffic going to or coming from outside the NETWARS architecture. The study lead must have an approach on how to address this issue.
4. The imported traffic may also contain the Critical/C2 traffic, meaning the critical traffic may be represented twice. If it is significant enough to invalidate the results the Study Lead will need to develop an approach on how to separate the Critical traffic out of the probe data. One approach is to understand all the traffic going to a particular IP addresses. For example, if Common Operating Picture (COP) traffic is the only traffic going to and coming from a particular address and Critical IERs have been developed for COP, then the imported background traffic for that address can be removed.

If the study lead cannot resolve these issues then it may be more appropriate to develop background traffic, where the links of interest are loaded in an incremental manner (e.g., 10%, 20%, 30%, and 40%).

NETWARS allows analysts to simulate background traffic as an analytical approximation using the "Background Load" attribute on the critical links. The analytical background traffic is not an exact representation of every byte associated with the traffic, but it does accurately impact network and system utilizations. This enables the simulation to generate realistic latency and throughput results for the critical traffic. Therefore, the impact of the simulation validity would be minimal when compared to the improvements in runtime efficiency.

3.3.3.1 Background Traffic through MVI

In addition to the topology development capabilities described in Section 3.2.4, the MVI feature allows analysts to import network traffic data from systems like Cisco NetFlow or Concord eHealth. The importation mechanism generates corresponding traffic demands based on the data in the associated network management files. The process can greatly enhance the validity of the background traffic representation and can greatly improve the efficiency of the process. More information regarding the details of the process can be found in the NETWARS Users Manual and in OPNET Technologies MVI Use Case documentation.

3.3.4 Network Probes

Network probes are devices that collect real-time or near real-time samplings of data traffic across network links. The analyst should review network probe data for consistency with study objectives. The following three characteristics should help the analyst identify useful network probe data.

- **Aggregation resolution** – Network probes record every possible packet for each source/destination pair across a link or interface. This packet data is accumulated into user-defined intervals. To generate background traffic, the analyst requires probe data at intervals of sufficient resolution.
- **Coverage** – The ability to configure network probes throughout an entire architecture is often inefficient and beyond the scope of the study. Therefore the analyst should work with the study lead to identify critical links or locations with the study architecture. The information can be abstracted to represent the "typical" background traffic across the entire network.
- **IP-Circuit Mappings** – The network probe data defines the producer and consumer pair through IP addresses. This does not allow the analyst to easily map probe data to associated circuits or links in the network unless he/she knows the IP addresses of the entire network. If the analyst can secure this information, the probe data can be a valuable source of background and critical traffic. Background traffic generated from probe data with detailed IP circuit mappings allows the analyst to represent all network contention.

The probe data filenames encode the port probed; however, port number mapping is not available. The Government should provide the IP addresses of several key servers throughout the architecture. Once an association between IP addresses and servers is made, the probe data can be used to help validate the critical traffic collected for the study. This involves extracting the daily profile for a given server's outgoing data flow and checking against known products (e.g., critical IERs) to be delivered from that server at specific times. If such a correlation can be made, the results will also be used to subtract the critical traffic contribution from the background traffic load. This will help eliminate the double counting of critical traffic.

3.3.5 Network Management Data

Network management software systems typically have the ability to collect health and status information for network assets. Usually, network managers are not concerned with IP-level information and instead prefer to see snapshots of the overall system and its major links and nodes. Accordingly, network management systems provide the ability to capture network utilization statistics. The analyst should examine the sampling rate of the network management data in a manner similar to the examination of network probe data. Network management data often gives the analyst with the ability to represent network contention at the higher levels of the architecture. Therefore, studies that require detailed and realistic local area network (LAN) contention will be limited by network management data.

3.4 SCENARIO VERIFICATION

The NETWARS analyst should always perform some amount of verification during the implementation of the scenario and before the execution of the study. The verification process is a systematic approach to finding defects in the scenario requirements and design. Especially promising is the potential for the early detection of subtle and potentially costly defects that are not identified even in extensive testing. The scenario verification relies on the architecture plans, the study scenario, the traffic, and the prerelease testing activities. To ensure a timely execution of verification activities, the analyst should define the criteria that correlate to a successful scenario.

3.4.1 Scenario Architecture

Because of the number of user inputs needed to develop a scenario, the process can contain many unintended user errors. These errors occur during the development of scenarios within the Scenario Builder. To minimize the occurrence of errors, the analyst should compare NETWARS scenario components with the critical design and check for consistency. The NETWARS analyst should also incorporate inspections performed by other study members.

3.4.2 Simulation Representation

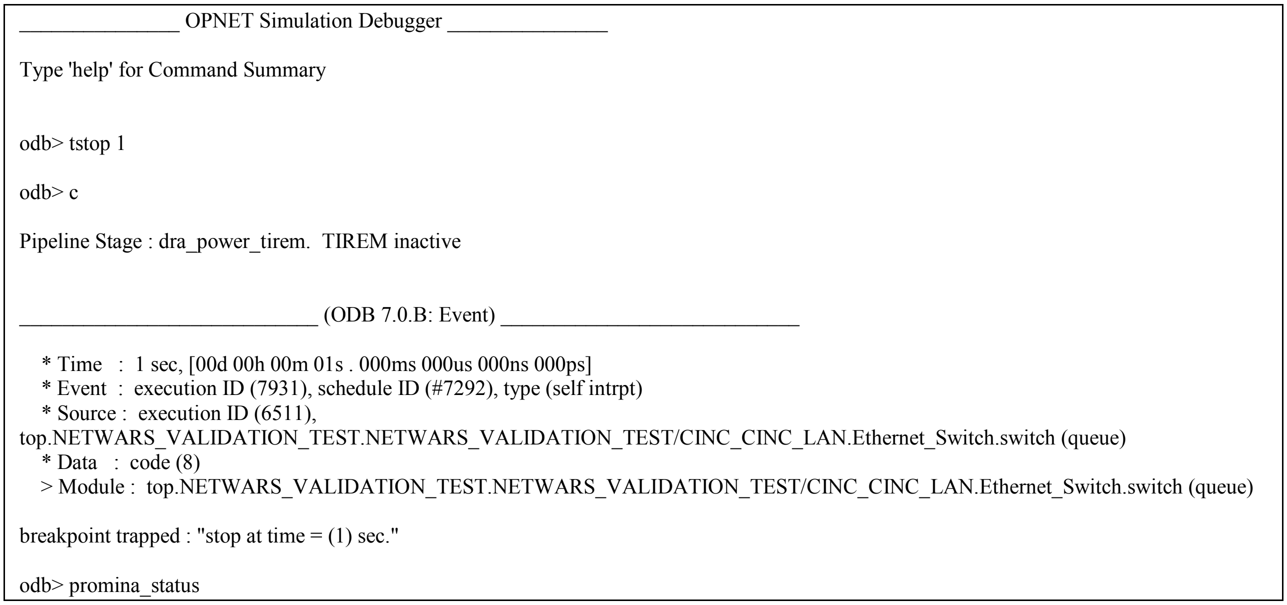
The NETWARS analyst should rely on the Simulation Domain to identify configuration errors throughout the development process and to ensure the integrity of the simulation results.

The simulation log is a powerful tool for the analyst to identify unexpected or unusual simulation behavior that might impact the integrity of the simulation results. A simulation log entry is generated automatically by certain simulation events. Each simulation log entry lists the simulation time, the associated simulation nodes, the symptoms surrounding the event, possible reasons for the event, and suggestions to eliminate the event. Appendix B lists some of the most commons simulation log entries.

3.4.2.1 Model Traces

The analyst should rely on model traces to verify the configuration within the Simulation Domain. The debugging environment for a NETWARS simulation is identical to the OPNET odb-debugging environment. To trigger the debugging environment in the Simulation Domain, the analyst should choose the "Use OPNET Simulation Debugger (ODB)" option inside the "Configure/Run Discrete Event Simulation" pull down menu. Section 3.5.1 describes this file and other simulation attributes in greater detail.

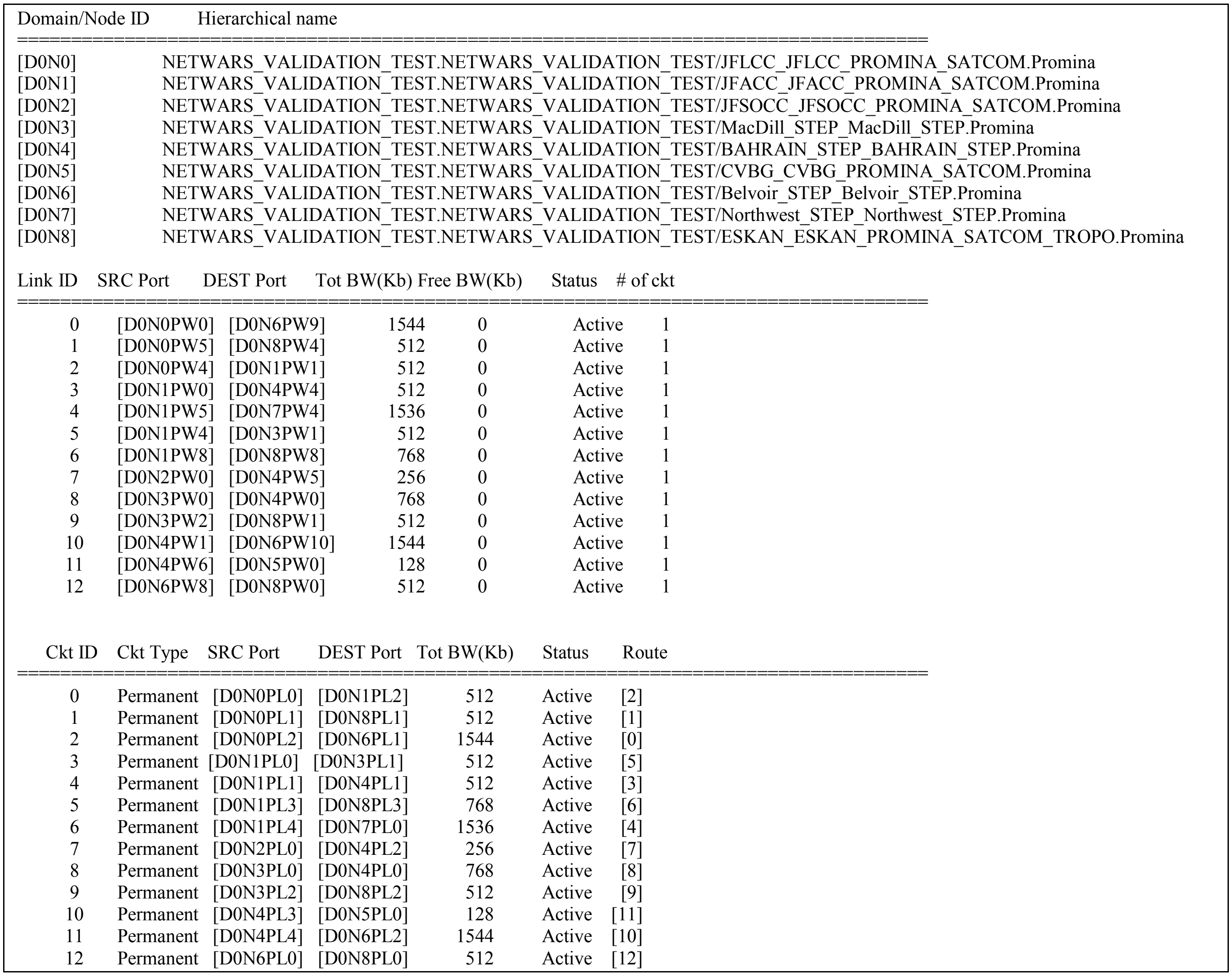
Analysts attempting to verify Promina architectures should use the Promina status trace built into the Promina model. This trace outputs the name, domain, and node ID for each Promina, the status of all WAN or Promina-to-Promina links, and the status of each Promina "circuit." The status information can help the analyst identify configuration errors that may impact performance. To execute the Promina trace, the analyst should establish a simulation time stop at 1 second and continue the simulation execution until that time stop. Once the simulation has stopped at the user-defined time stop, the analyst should trigger the Promina status trace. Figure 2 illustrates an example of executing the Promina status trace.



```
______________ OPNET Simulation Debugger ______________

Type 'help' for Command Summary


odb> tstop 1

odb> c

Pipeline Stage : dra_power_tirem.  TIREM inactive


__________________________ (ODB 7.0.B: Event) __________________________

   * Time   :  1 sec, [00d 00h 00m 01s . 000ms 000us 000ns 000ps]
   * Event  :  execution ID (7931), schedule ID (#7292), type (self intrpt)
   * Source :  execution ID (6511),
top.NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/CINC_CINC_LAN.Ethernet_Switch.switch (queue)
   * Data   :  code (8)
   > Module :  top.NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/CINC_CINC_LAN.Ethernet_Switch.switch (queue)

breakpoint trapped : "stop at time = (1) sec."

odb> promina_status
```

Figure 2 - Executing Promina Status Trace

Figure 3 exhibits the results of a sample execution of the Promina status trace.



```
Domain/Node ID        Hierarchical name
==========================================================================================
[D0N0]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/JFLCC_JFLCC_PROMINA_SATCOM.Promina
[D0N1]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/JFACC_JFACC_PROMINA_SATCOM.Promina
[D0N2]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/JFSOCC_JFSOCC_PROMINA_SATCOM.Promina
[D0N3]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/MacDill_STEP_MacDill_STEP.Promina
[D0N4]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/BAHRAIN_STEP_BAHRAIN_STEP.Promina
[D0N5]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/CVBG_CVBG_PROMINA_SATCOM.Promina
[D0N6]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/Belvoir_STEP_Belvoir_STEP.Promina
[D0N7]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/Northwest_STEP_Northwest_STEP.Promina
[D0N8]          NETWARS_VALIDATION_TEST.NETWARS_VALIDATION_TEST/ESKAN_ESKAN_PROMINA_SATCOM_TROPO.Promina

Link ID   SRC Port     DEST Port    Tot BW(Kb) Free BW(Kb)    Status   # of ckt
==========================================================================================
     0    [D0N0PW0]  [D0N6PW9]       1544      0          Active      1
     1    [D0N0PW5]  [D0N8PW4]        512      0          Active      1
     2    [D0N0PW4]  [D0N1PW1]        512      0          Active      1
     3    [D0N1PW0]  [D0N4PW4]        512      0          Active      1
     4    [D0N1PW5]  [D0N7PW4]       1536      0          Active      1
     5    [D0N1PW4]  [D0N3PW1]        512      0          Active      1
     6    [D0N1PW8]  [D0N8PW8]        768      0          Active      1
     7    [D0N2PW0]  [D0N4PW5]        256      0          Active      1
     8    [D0N3PW0]  [D0N4PW0]        768      0          Active      1
     9    [D0N3PW2]  [D0N8PW1]        512      0          Active      1
    10    [D0N4PW1]  [D0N6PW10]      1544      0          Active      1
    11    [D0N4PW6]  [D0N5PW0]        128      0          Active      1
    12    [D0N6PW8]  [D0N8PW0]        512      0          Active      1


   Ckt ID   Ckt Type   SRC Port      DEST Port   Tot BW(Kb)     Status     Route
==========================================================================================
     0    Permanent  [D0N0PL0]  [D0N1PL2]        512      Active     [2]
     1    Permanent  [D0N0PL1]  [D0N8PL1]        512      Active     [1]
     2    Permanent  [D0N0PL2]  [D0N6PL1]       1544      Active     [0]
     3    Permanent [D0N1PL0]  [D0N3PL1]         512      Active     [5]
     4    Permanent  [D0N1PL1]  [D0N4PL1]        512      Active     [3]
     5    Permanent  [D0N1PL3]  [D0N8PL3]        768      Active     [6]
     6    Permanent  [D0N1PL4]  [D0N7PL0]       1536      Active     [4]
     7    Permanent  [D0N2PL0]  [D0N4PL2]        256      Active     [7]
     8    Permanent  [D0N3PL0]  [D0N4PL0]        768      Active     [8]
     9    Permanent  [D0N3PL2]  [D0N8PL2]        512      Active     [9]
    10    Permanent  [D0N4PL3]  [D0N5PL0]        128      Active    [11]
    11    Permanent  [D0N4PL4]  [D0N6PL2]       1544      Active    [10]
    12    Permanent  [D0N6PL0]  [D0N8PL0]        512      Active    [12]
```

Figure 3 - Sample Promina Status Trace

3.4.3 Traffic

The multiple methods for defining or associating IERs with a given scenario can cause the analyst confusion and lead to user error. Specifically, the analyst often encounters user errors in the IER text files. Therefore, the verification of all scenario IERs is a critical step before the execution of study runs. The analyst should use the export IER report utility provided in the Scenario Builder to generate a tab delimited text report that lists all IERs associated with a given scenario. This file allows the analyst to verify the inclusion of each IER and also to verify the exact characteristics of each IER.

The analyst should also verify the ability of each IER producer/consumer pair in a given study to send a dummy IER. To accomplish this check, the analyst should develop a unique test IER text file with an IER for each producer/consumer pair. To avoid contention on the network, the analyst should use the start and stop times to ensure that each IER is sent individually. This check often identifies network configuration errors that are difficult to isolate under normal traffic loads.

3.5 EXECUTING NETWARS

Before running a simulation the analyst should first clearly define the study objective. Defining a study objective will make it much easier to then decide on the parameters of each simulation that needs to be run.

3.5.1 NETWARS Simulations

The NETWARS analyst should develop and maintain a simulation run matrix based on study objectives and simulation characteristics. The run matrix should encompass all study excursions and should identify key simulation characteristics such as scenario duration and associated seeds. The analyst should understand the requirements associated with the analysis approach and ensure that the run matrix fulfills those requirements.

3.5.1.1 Simulation Attributes

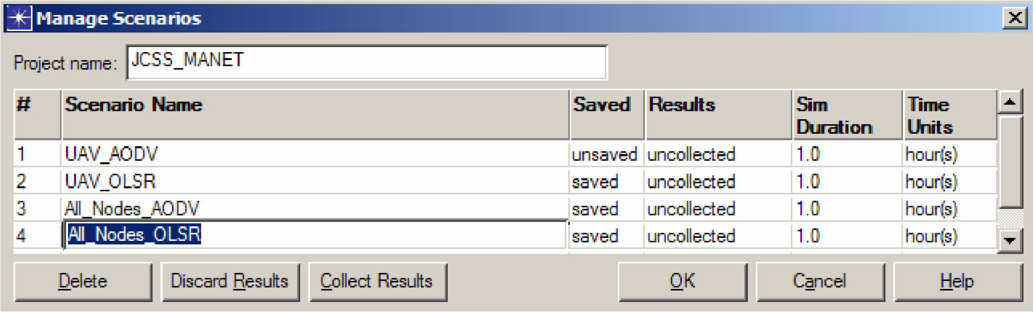
NETWARS allows users to configure simulation attributes using the “Configure/Run DES” menu. This gives users greater visibility into the different simulation attributes and allows users to optimize the scenario, in terms of simulation run time. The following is a list of some of the simulation attributes.

- **"OSPF Sim Efficiency":** Allows the analyst to control whether OSPF continues to be active beyond a certain time in the simulation. Specifically, OSPF will no longer send “hello” packets beyond the time specified in the “OSPF Stop Time” attribute, which is measured in seconds of simulated time. “Hello” packets are keep-alive packets used to determine if topology changes have occurred. If no topological changes are anticipated (e.g., failure and recovery excursions), the “OSPF Sim Efficiency” attribute should be one (1), and the “OSPF Stop Time” should be set to a reasonable time frame. Generally, a value equal to three (3) times the “OSPF Routing Table Interval” performs well. The “OPSF Routing Table Interval” is by default set to 60 seconds; this means that the first routing table calculations are performed at 60 seconds. Therefore, application traffic running over IP in the presence of OSPF should not begin generating traffic until after this time.
- **"Eth Hub Optimization":** Enables/disables the optimization flag for an Ethernet hub. If the optimization flag is disabled, the hub will send a copy of every packet it receives without collision to every medium access control (MAC) that is connected to it. If enabled, the hub will forward the packet only to its destination MAC. If this MAC is not one of the hub's clients, the hub will forward the packet to all of its client MAC addresses that belong to a bridge or switch.

- **“IP Routing Protocol":** Specifies the routing protocols used on the interface. By default, this attribute is set to RIP for all interfaces in the network.
- **“ATM Dynamic Routing Protocol":** Determines the routing protocol used by the asynchronous transfer mode (ATM) switches. When set to Distance-Vector, all switches in the network use the Distance-Vector Routing Protocol. Using the Private Network-to-Network Interface (PNNI) setting reduces simulation run time by using the static PNNI routing protocol.
- **“MSE Hello Interval":** Specifies the amount of time in seconds between mobile subscriber equipment (MSE) hello messages. The MSE systems use this message to detect system and/or link failure recovery.
- **“Promina Hello Interval":** Specifies the amount of time in seconds between Promina hello messages. The Promina systems use this message to detect system and/or link failure and recovery.
- **“Promina Packet Segment Size"**: Specifies the maximum Promina frame size. Promina frames are system-specific frames that are sent between Promina trunk or WAN interfaces.
- **“Error Calculation for Cosite Interference":** Flagged to skip the cosite interference calculation in the OPNET standard wireless pipeline process. The default value of this attribute enables this calculation; otherwise this value is set to zero (0), which means the cosite interference is not modeled.
- **“Tracer Packets Per Interval":** Used to calculate the background traffic delay statistic by specifying the number of tracer packets sent for a given period of constant use.
- **“Crypto Synchronization On/Off”:** In NETWARS the encryption devices (e.g., KIV-7 and KG-195) periodically resynchronize, which is determined by the value of the synchronization_time attribute. The Crypto Synchronization On/Off (0=off/1= on) attribute enables/disables the crypto resynchronization feature.
- **“EIGRP Sim Efficiency":** When enabled, all EIGRP activities stop after the EIGRP Stop Time value is reached. Enabling this attribute makes the simulation more event- and execution-time efficient but does not model EIGRP consequences of topology changes (e.g., failure and recovery excursions) that occur after the EIGRP Stop Time. If no topological changes are anticipated, the “EIGRP Sim Efficiency” attribute should be one (1) and the “EIGRP Stop Time” should be set to a reasonable time frame.
- **“EIGRP Stop Time":** Specifies the EIGRP Stop Time used when the EIGRP Sim Efficiency attribute is enabled. This attribute should never be less than the amount of time required to establish the routing tables (approximately 100 seconds)

3.5.1.2 Batch Simulation of Scenarios inside a NETWARS Project

To reduce the time required to execute multiple simulations for each scenario of a project, the analyst should use the ‘*Manage Scenarios’* option. The following are step-by-step instructions to execute a batch simulation of NETWARS scenarios:



Project name: JCSS_MANET

#	Scenario Name	Saved	Results	Sim Duration	Time Units
1	UAV_AODV	unsaved	uncollected	1.0	hour(s)
2	UAV_OLSR	saved	uncollected	1.0	hour(s)
3	All_Nodes_AODV	saved	uncollected	1.0	hour(s)
4	All_Nodes_OLSR	saved	uncollected	1.0	hour(s)

Manage Scenarios

1. Select: Scenarios > Manage Scenarios
2. Set the appropriate simulation duration (in seconds) for each scenario in the project.
3. For each scenario you want to simulate, select "Collect Results" from the "Results" cell.
4. Click OK to close the Manage Scenarios dialog box and begin running the simulations.

A simulation dialog box appears for the first scenario, showing the progress of the simulation. Additional scenarios are simulated serially in the order shown in the Manage Scenarios dialog box.

The '*Manage Scenario'* option also gives the user an option to delete a scenario.

3.5.1.3 Batch Simulations of Different NETWARS Projects

While the *'Manage Scenarios'* GUI allows simulations to be conducted for multiple scenarios within a single project, in some cases it may be convenient to run simulations for multiple scenarios across multiple project files. This can be accomplished by creating a batch file, for the scenarios that you want to simulate, and executing the file from the NETWARS Console. In addition to creating a batch file, several more steps must be taken in order to successfully configure and execute a batch simulation. These steps are described in detail below.

3.5.1.3.1 Creation/Manipulation of the scenario environment file (.ef file)

In order to successfully include a scenario as part of a batch simulation, an environment file must already exist in the project folder. The environment file contains the values of simulation attributes that are normally configured through the DES GUI. The environment file is generated for a scenario once it is executed in a simulation. For this reason, batch simulations may prove to be most valuable when it becomes necessary to rerun simulations for a large number of scenarios across different projects.

In the event that batch simulation is the desired means of conducting multiple simulations for scenarios that have not yet been run, the environment files can be created by running a simulation for each scenario through the GUI for a duration of one second. After the simulation is run for one second and the environment files have been generated, the duration attribute can be changed in the environment files themselves. To do this, open up each environment file in notepad and find the duration attribute. At this point, the attribute will be set to a value of 1 since the simulation that you conducted was run for one second. Manually change the value of the duration attribute to whatever duration (in seconds) is desired for the scenario in your batch simulation.

3.5.1.3.2 Modify the env_db12.0 file

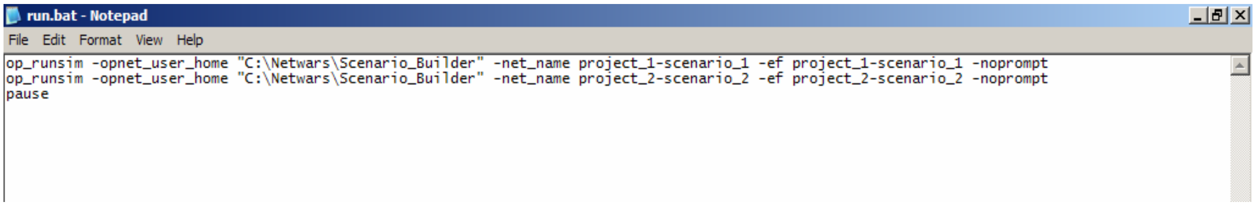
The env_db file is a text file that contains the values of all of the advanced preference attributes in NETWARS. The file is located in the Netwars\Scenario Builder\op_admin directory.

In order to include scenarios in a batch simulation, the complete path of each scenario folder must be added to the mod_dirs attribute in the env_db file. Adding the correct scenario directory allows the .xml traffic file to be accessed during the batch simulation. The syntax for all of the existing model directory paths must be followed when adding the paths for your scenario folders.

It is important to note that env_db file overwrites itself each time NETWARS is opened. Therefore, once the scenario directories have been added to the env_db file, NETWARS should not be reopened until the batch simulation has been successfully executed. If NETWARS has to be opened before the batch simulation has executed, the env_db file can be saved to a different directory and saved back to its original location when the batch is ready to be executed.

3.5.1.3.3 Create the batch file (.bat file)

A batch file is a file with a .bat extension that contains a sequence of commands and file names that define which scenarios will be included in a batch simulation. To create a batch file, first create a text file that contains the op_runsim command along with the input parameters that define what scenarios to include in the batch simulation. Next, save the file with a .bat extension. The syntax needed to include two scenarios from two different projects in the batch file is shown in Figure 4 below.



```
op_runsim -opnet_user_home "C:\Netwars\Scenario_Builder" -net_name project_1-scenario_1 -ef project_1-scenario_1 -noprompt
op_runsim -opnet_user_home "C:\Netwars\Scenario_Builder" -net_name project_2-scenario_2 -ef project_2-scenario_2 -noprompt
pause
```

Figure 4 – Sample Batch File

The syntax shown for each of the two scenarios above is required for every scenario that you want to include in the batch simulation. The op_runsim command executes a simulation for the parameters specified on that line of the file. In this case, those parameters are

-opnet_user_home, –net_name, and –ef. The value of the –opnet_user_home parameter is Netwars\Scenario_Builder. The values of the –net_name and –ef parameters are the names of the network model (nt.m file) and environment file (.ef file) respectively. The names of those two files follow the convention project_name-scenario_name by default. The three parameters described here are just a few of many parameters that may be included in a batch file. Typing the command –noprompt at the end of each line causes the batch simulation to continue to execute without pausing and prompting the user for parameters that are not included in the batch file. On the last line of the batch file, type the command *'pause'*. This will cause the console window to remain open until it is closed by the user.

3.5.1.3.4 Executing the batch simulation

Once all of the necessary files have been created and configured, the batch simulation can be executed. First, select Start > Programs > Netwars > Netwars Console from the Start menu. Next, drag the .bat file that you created and drop it inside the console; press Enter to start the batch simulation. Any errors that occur during the execution of the batch simulation will generate a message that will appear in the console. Once the batch simulation is complete, collected statistics can be viewed by opening each scenario and viewing the results in Results Analyzer.

3.5.2 Simulation Efficiency

The efficiency of a simulation also can factor into the determination of a study run matrix. Because NETWARS uses a discrete event Simulation Engine, simulation run times can vary greatly depending upon the size of the scenario, the type of scenario, the amount of traffic, and the type of traffic. To help analysts understand and predict run times, the baseline scenario with and without traffic should be executed to establish expectations. The analyst can, if applicable, reduce the run time by decreasing the simulation duration or decreasing the scenario complexity. Any changes should be discussed with all study members to reach a consensus. The following list of actions employed by past study efforts can improve simulation efficiency:

- Turn on EIGRP, OSPF, and ATM simulation efficiency mode. (see Section 3.5.1.1)
- Use the analytical background traffic methodology. (see Section 3.3.3.3)
- Adjust Promina frame size to a larger value. (see Section 3.5.1.1)
- Adjust the satellite terminal frame size for each OPFAC with a satellite terminal.
- Aggregate LAN components such as workstations, Ethernet hub, and IP routers to reduce the number of devices.
- Import IP routing table from earlier simulations to reduce initialization time.

3.5.3 Utilizing TIREM

The NETWARS Simulation Domain can utilize the Terrain-Integrated Rough-Earth Model (TIREM) for wireless propagation calculations. TIREM is a DoD application that includes the effects of terrain on communications between wireless transmitters and receivers.

If you wish to replace TIREM 3.14 with TIREM 3.15 in NETWARS, you need the following information.

The following files are directly related to the functionality of TIREM in NETWARS:

TIREM 3.14	TIREM 3.15
TIREM314.dll	TIREM315DLL.dll
TIREM314.lib	TIREM315DLL.lib
WOTRet205DLL.dll	WOTRet207DLL.dll
WOTRet205DLL.lib	WOTRet207DLL.lib
tirem_support.ex.c (3/2/02)	DFORRT.dll
	tirem_support.ex.c (9/19/02)

TIREM314.dll and TIREM314.lib contain code used to compute propagation losses for NETWARS. WOTRet205DLL.dll and WOTRet205DLL.lib contain code used to obtain terrain profiles from WOTL files created from DTED data. These TIREM 3.14 dll and lib files are located in the \NETWARS\Scenario_Builder\11.5.A\netwars\bin and \NETWARS\Sim_Domain\op_models\netwars\std_models\misc\tirem folders. These files should be replaced by their TIREM 3.15 equivalents in the same folders. The equivalent files are TIREM315DLL.dll, TIREM315DLL.lib, WOTRet207DLL.dll, and WOTRet207Dll.lib. Since the TIREM 3.15 files have different names, there is no need to delete the TIREM 3.14 files.

The tirem_support.ex.c file contains C code that loads and uses the TIREM 3.14 files. This file is located in the \NETWARS\Sim_Domain\op_models\netwars\std_models\misc\tirem folder. It should be replaced by the version of the tirem_support.ex.c that references the TIREM 3.15 files. In addition the tirem_support.iO.ex.o file should be deleted, to force NETWARS to compile the new version of tirem_support.ex.c file.

The TIREM315DLL.dll file references the DFORRT.dll. It should be placed in a folder such as \Winnt\system32\ where NETWARS can find it.

The locations of the WOTL files are specified by another file called topoman.ini. This file should be located in the \Winnt folder. The topoman.ini file should contain the path to the folder containing the topo.inf file and the path to the folder containing the Gene.dat, Gene.ddr, and Gene.ddt files. The WOTL files should be located in the folders specified in the topoman.ini file before running the new UserGuide_TI_Scenario.nt.m with either TIREM 3.14 or TIREM 3.15.

3.6 ANALYZING NETWARS RESULTS

The analysis of NETWARS results is the critical responsibility for the analyst. The key to this process is the development of a robust analysis approach that is consistent with study goals and objectives. A NETWARS analysis approach will rely on the evaluation and comparison of measures of performance (MOP) generated by the simulation.

3.6.1 Analysis Approach

Each NETWARS simulation generates a significant amount of simulation result data that requires a structured and disciplined analysis approach. The analyst must develop a clear understanding of the relationship between simulation performance and the study goals/objectives. Most NETWARS studies will contain a baseline configuration with a set of excursions that allow the analyst to evaluate and assess a wide variety of system or network changes or variations. The following is a short list of some of these changes or variations:

- Integration of new communications systems or technologies
- Upgraded network and/or system capacity
- Increased or decreased network traffic
- Additional mission-critical applications
- Network and/or system failure.

The analyst can use a wide variety of simulation results or MOPs to assess and analyze each study variation. NETWARS generates a robust set of MOPs that correspond to network, system, and traffic performance. Sections 3.6.2 and 3.6.3.1 describe all of the NETWARS statistics.

3.6.1.1 Expectations

Before launching a set of simulations, the analyst should use quick analytical techniques to evaluate the network, link, and system utilization associated with a given traffic load. This process can help the analyst identify network and system bottlenecks that could significantly impact the simulation results. The analyst should use the evaluation feature in the NETWARS Capacity Planner to help evaluate network-wide performance. The NETWARS analyst should also develop expected traffic loads between producer/consumer pairs. This process can help the analyst predict the “busy” talkers in the network. Often the busy talkers tend to experience the most significant variations in network performance.

3.6.1.2 Collecting Statistics

MOPs are the most typical NETWARS analytical product that relate to either network-wide and application performance or individual-device performance. The analyst should define an initial set of MOPs based on the performance expectations generated through analytical-based analysis. Throughout the simulation execution phase, the analyst should adjust the collection strategy based on earlier simulation results. Figure 5 shows the process of tailoring the MOP collection strategy on the basis of simulation results.

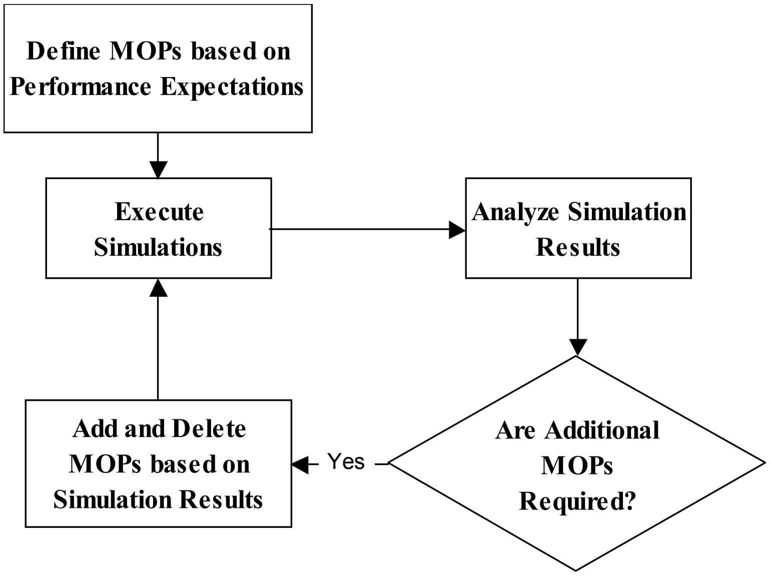

Define MOPs based on Performance Expectations
Execute Simulations
Analyze Simulation Results
Are Additional MOPs Required?
Yes
Add and Delete MOPs based on Simulation Results

Figure 5 - Determine Simulation Statistics

The network-wide MOPs such as link utilization often provide a high-level view of performance. The study lead can assume that the lower the utilization, the more stable and flexible the network architecture. Application or IER MOPs such as end-to-end latency often provide the study lead with the empirical data behind a measure of effectiveness (MOE). The empirical data associated with IER performance combined with user expectations can be a powerful tool for the study lead. For example, if the analyst defines a maximum delay for an IER, to ensure task or mission effectiveness, MOPs can be used to draw conclusions regarding the impact of network events such as outages or architectural changes on mission effectiveness.

3.6.1.3 Statistical Variability

Statistical variance can play a key role in determining the number of runs necessary for a study. NETWARS scenarios rely on the random number seed generator to provide statistical variance in the generation of IERs. The analyst should establish confidence intervals expectations for each study MOP and predict the MOP sample size associated with each simulation execution. The sample size helps the analyst determine the number of simulations, with different random seeds required to produce an acceptable MOP confidence interval. By default, the analyst should execute at least three different seeds for each excursion. Once the first three seeds have been analyzed, the number of additional simulations necessary to achieve a desired stability for each MOP must be determined.

3.6.2 Simulation Output

The NETWARS simulation generates text-based output files for simulation results. The simulation files fall into two distinct categories: IER/thread-related statistics and Output Vector (OV)-based statistics. The same set of IER/thread-related statistics is generated for each simulation. The OV-based statistics are the user-defined device and link statistics. These statistics are collected with an OPNET standard binary format called an OV file.

3.6.2.1 IER/Thread Statistic Files

The text files used for writing IER statistic information have the following naming convention -- <scenario_name>.ier_fail, <scenario_name>.ier_sent, and so forth. These files are located in the following directory: <root directory>/netwars/user_data/projects/<project name>\<project phase>\results\. The first row in each file is the heading row. It has the names of all the fields, separated by tabs. All subsequent rows contain the information about IERs.

Table 4 – <scenario_name>.ier_sent format

Field name	Description
IER_ID	A unique identifier for each IER (integer)
Th_ID	A unique identifier for each thread (integer)
IER_Src_Pf	The OPFAC that generates this IER
IER_Dest_Pf	The OPFAC that acts as the consumer for this IER
IER_Type	Specifies the traffic type (Voice, Data)
IER_Class	Specifies the security classification (Unclassified, Secret, Confidential, etc.)
IER_Size	Specifies the size of the IER. Number of bytes for data IERs and duration (in seconds) for voice IERs.
IER_Start	Time at which IER transmission was first attempted
IER_Sent	Time at which IER was actually transmitted
SE_Over	Name of the consumer end-system device
Blocks	Number of times the IER was blocked before transmission
IER_Priority	Specifies the precedence (Routine, Immediate, Priority, Flash, Flash Override)

Table 5 – <scenario_name>.ier_rcvd format

Field name	Description
IER_ID	A unique identifier for each IER (integer)
IER_Src_Pf	The OPFAC that generates this IER
IER_Dest_Pf	The OPFAC that acts as the consumer for this IER
IER_Type	Specifies the traffic type (Voice, Data)
IER_Start	Time at which IER transmission was first attempted
IER_Rcvd	Time at which IER was received
Th_ID	A unique identifier for each thread (integer)
IER_Class (security classification)	Specifies the security classification (Unclassified, Secret, Confidential, etc.)
IER_Perish	The user-defined IER perishability limit
IER_Priority	Specifies the precedence (Routine, Immediate, Priority, Flash, Flash Override)
IER_Desc	Description of this IER
IER_Size	Specifies the size of the IER. Number of bytes for data IERs and duration (in seconds) for voice IERs

Table 6 – <scenario_name>.ier_block format

Field name	Description
IER_ID	A unique identifier for each IER (integer)
Th_ID	A unique identifier for each thread (integer)
IER_Src_Pf	The OPFAC that generates this IER
IER_Type	Specifies the traffic type (Voice, Data)
IER_Start	Time at which IER transmission was first attempted
Block_Time	Time at which the IER blocked
Block_Reason	Reason for IER blocking

Table 7 – <scenario_name>.ier_fail format

Field name	Description
IER_ID	A unique identifier for each IER (integer)
Th_ID	A unique identifier for each thread (integer)
IER_Src_Pf	The OPFAC that generates this IER
IER_Dest_Pf	The OPFAC that acts as the consumer for this IER
IER_Type	Specifies the traffic type (Voice, Data)
IER_Start	Time at which IER transmission was first attempted
Fail_time	Time at which the IER failed
Fail_Reason	Reason for IER failure

Similar to the IER statistic files, the text files used for writing thread statistic information have the following naming convention -- <scenario_name>.th_fail, <scenario_name>.th_sent, and so forth. These files are located in the following directory: <root directory>/netwars/user_data/projects/<project name>\<project phase>\results\. The first row in each file is the heading row. It has the names of all the fields, separated by tabs. All subsequent rows contain the information about threads.

Table 8 – <scenario_name>.th_sent format

Field name	Description
Thd_ID	A unique identifier for each thread (integer)
Thd_Instance	Instance number of the thread firing
Src_platform	The OPFAC that generates this thread
Rxn_Criticality	Information whether the reaction IER is critical or not
Rxn_IER_ID	IER ID for the reaction IER
Rxn_Start	Time when the reaction IER is fired
Thd_Start_Time	Time at which thread transmission was started
Thd_Stop_Time	Time at which the thread transmission stopped
Thd_Distribution	Distribution for the thread interarrivals
Thd_Interarrival	Interarrival times for the threads

Table 9 – <scenario_name>.th_rcvd format

Field name	Description
Thd_ID	A unique identifier for each thread (integer)
Thd_Instance	Particular instance firing of the thread
Src_platform	The OPFAC that generates this thread
Dest_platform	The OPFAC that acts as the consumer for this thread
Rxn_Criticality	Information whether the reaction IER is critical or not
Rxn_IER_ID	The IER ID of the reaction IER
Thd_Start	Time at which thread transmission was first attempted
Thd_End	Time at which thread transmission was received at destination

Table 10 – <scenario_name>.th_fail format

Field name	Description
Th_ID	A unique identifier for each thread (integer)
Th_Start	Time at which thread transmission was first attempted
Th_Fail	Time at which thread transmission was determined to have failed
Th_Src_Pf	The OPFAC that generates this thread
Th_Fail_Pf	The OPFAC that acts as the consumer for this thread
Fail_IER_ID	The ID of the thread segment at which the thread failed
Fail_time	Time at which the thread failed
Fail_Reason	Reason for thread failure

3.6.2.2 OV-Based Statistics

The CDMs and link statistics utilize OPNET standard vector statistics. The analyst defines the capture methodology for these types of statistics through the NETWARS OPFAC attribute window and the NETWARS link attribute window. The simulation collects these statistics in a custom binary format called an OV file. The following tables describe the different link and broadcast statistics.

Table 11 – Link Statistics Files

Statistic name	Comments
voice_throughput	Records continuous stream of 16 Kbps packets for each call for the duration of the call
data_throughput	Records the size and the duration of data (data IERs & voice signaling packets) packets on the inter-OPFAC links. This statistic is collected in both forward and reverse directions. The naming convention used is: <hierarchical name of link>__<hierarchical name of device1>__<hierarchical name of device2>__in the direction from device1 to device2.
channel_percentage_utilization	Voice throughput as the percentage of the total link bandwidth

Table 12 – Broadcast Network Statistic Files

Statistic name	Comments
broadcast_network_utilization	Recorded by radios participating in a broadcast radio network

3.6.3 Measures of Performance

The following subsections describe the different MOPs available in NETWARS. The analyst should realize that all of these metrics are based on the text-based files generated by the simulation. Therefore, if the performance or the flexibility of NETWARS does not meet the requirements of the study, the analyst can develop similar metrics through Microsoft Excel or other spreadsheet applications.

3.6.3.1 IER-Related MOPs

NETWARS allows the analyst to examine the IER-related MOPs in two different manners: global and OPFAC-specific. The global option allows the analyst to examine the performance of all IERs of a certain traffic type or priority. The OPFAC-specific statistics allow the user to examine the performance of the IERs that were either sent or received by a specific OPFAC. Table 13 lists the different IER-related MOPs.

Table 13 – IER-Related MOPs

MOP	Description
IERs sent count	A cumulative count of IERs sent
IERs rcvd count	A cumulative count of IERs received
IER Failed	A cumulative count of IERs failed
IER Perished	A cumulative count of IERs perished
Connection Latency	The latency (IER sent time – IER start time) in establishing a connection
Speed of Service	The delay (IER received time – IER start time) computed for each received IER
Message Completion Rate	The ratio of the number of data IERs received to the number of data IERs sent
Message Error Rate	The ratio of the number of data IERs that failed to the number of data IERs sent
Call Completion Rate	The ratio of the number of voice IERs received to the number of voice IERs sent
Number of Blocks for each IER Sent	The number of times each IER was blocked
Blocking Probability	The ratio of the number of IERs that were blocked at least once to the number of IERs sent
Grade of Service	The percentage of IERs received within the perishability limitation associated with each IER
Perishability for the Rcvd IERs	A cumulative count of IERs received where the delay (IER received time – IER start time) is greater than the perishability limitation associated with each IER

3.6.3.2 Link-Related MOPs

NETWARS also allows the analyst to examine link-related MOPs. The content of link MOPs is directly related to the capture methodology defined for each link. The analyst can either capture all values or employ a bucket mode that captures one hundred (100) time-based averages equally distributed throughout the complete simulation time. The analyst should ensure that the capture methodology is consistent with the level of detail required by the analysis approach. The following table lists the different link-related MOPs.

Table 14 – Link-Related MOPs

MOP	Description
Channel Percent Utilization	The percentage of voice channels that are utilized
Data Throughput – Forward/Reverse (bits/sec)	The amount of data the traverses a link in a forward or reverse direction
Voice Throughput (bits/sec)	Since NETWARS voice calls utilize a user-defined amount of bandwidth (16kbps or 32kbps), the amount of bandwidth associated with voice calls that traverse a link

3.6.3.2.1 Device-Related MOPs

NETWARS also allows the analyst to examine the device-related MOPs. The content of device MOPs is directly related to the capture methodology defined for each OPFAC. The analyst can customize the capture mode to ensure that the capture methodology is consistent with the level of detail required by the analysis approach. NETWARS allows the analyst to capture a wide variety of device MOPs that are specific to each model. The following table lists the NETWARS standard device-related MOPs. Additional information regarding model-specific MOPs such as IP, ATM, and TCP statistics can be found in the OPNET ITGuru documentation set.

Table 15 – Device-Related MOPs

MOP	Description
Packets of Cells Dropped	The number of packets dropped by a network device
Throughput (bits/sec and packets/sec)	The amount of data that passes through a network device
Queue Size (bits or packets)	The size of the IP processing queue

3.7 QUALITY OF SERVICE (QOS)

QoS refers to the practice of categorizing network traffic and allocating different amounts of network resources based on the relative importance of each type of traffic. The goal of QoS is typically to provide improved or more predictable performance for high priority network traffic at the expense of network traffic with lower priority. Cisco describes three End-to-End QoS Models; Best Effort, Differentiated Service, and Integrated Service. Each of these service models represents a unique approach for providing the required level of service for network traffic from one end of the network to the other. Each service model implements a varied set of QoS capabilities, making each one appropriate for different types of network application traffic. Each service model is summarized below.

- **Best Effort:** Best effort service is the default configuration for most packet switched networks (i.e. the internet). In the absence of any explicitly configured QoS mechanisms within a network, all packets receive best effort treatment. When best effort service is implemented, an application may send data at any time, in any quantity, without receiving permission or notifying the network. The network delivers data if possible, without any assurance or performance guarantees.

- **Differentiated Service:** Differentiated service seeks to classify each data packet traversing a network and provide specific QoS according to the service class that each packet belongs to. Packets are most commonly classified using the ToS byte within the IP header or by using the source and destination IP address of the packet. Once packets are classified, they may receive differentiated service from any number of Diffserv QoS capabilities (i.e. Congestion Management, Congestion Avoidance, and Traffic Policing). A detailed description of the major QoS capabilities is provided in the next section.

- **Integrated Service:** Integrated service is similar to differentiated service in the sense that different QoS may be provided to different network traffic. However, unlike Diffserv, Intserv requires the application to request a particular level of service from the network prior to transmitting any data. This request is made in the form of explicit signaling in which the application provides the network with information regarding its traffic profile. Once the application receives confirmation from the network, data may be transmitted. The network commits to accommodating the required QoS for the application as long as the application transmits within the parameters of the profile that were established in the initial signaling.

The next section of the document will described the individual QoS capabilities that comprise each of the three service models summarized above. The section will primarily focus on Diffserv capabilities. Capabilities associated with the Best Effort and Intserv models will be described in a more cursory manner in the interest of completeness.

3.7.1 QOS Capabilities

In order to implement QoS mechanisms effectively in JCSS, the user must first have a basic understanding of each mechanism and the performance issue it is designed to address. The purpose of this section is to provide a common understanding of QoS; it is not intended to be an in depth discussion. This section will not present a comprehensive description of all QoS capabilities, but will instead focus on the major QoS capabilities that are currently available in JCSS.

3.7.1.1 Packet Classification

Packet classification is fundamental to the implementation and operation of QoS within a network. Packet classification provides the various QoS capabilities with a means of grouping packets into service classes and providing differentiated service to each class. The most common mechanism used to classify packets is the type of service (ToS) byte within the IPv4 packet header. A couple of different schemes are typically used to manipulate the bits within this field. These schemes are described in the subsections below.

3.7.1.1.1 ToS

The packet's priority is defined by manipulating the first three bits of the ToS byte. The contents of the ToS byte are illustrated below in Figure 6.

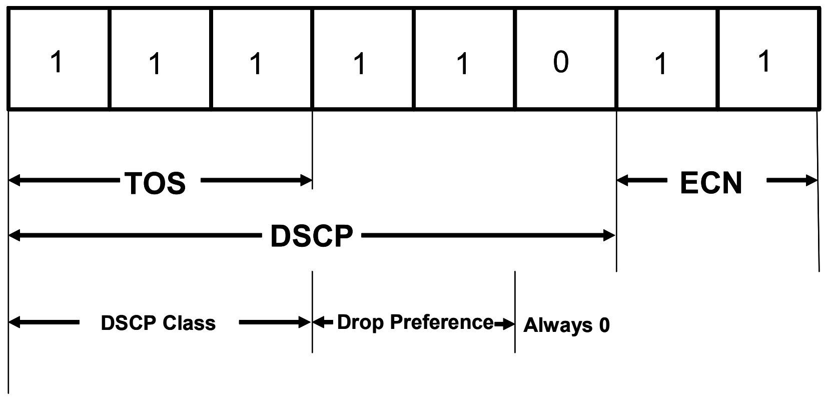


Figure 6
IPv4 TOS Byte

TOS uses only the first three bits, providing a maximum of eight different priorities:

- Reserved (7)
- Interactive Voice (6)
- Interactive Multimedia (5)
- Streaming Multimedia (4)
- Best Effort (3)
- Standard (2)
- Background (1)
- Best Effort (0)

3.7.1.1.2 Differentiated Services Code Point (DSCP)

DSCP uses the first six bits, providing up 63 different priorities (e.g., AF23). DSCP also has two parts:

- DSCP Class: Determines the priority of the application using the first three bits of the ToS byte (e.g., AF2)
- Drop Preference: Used to determine the drop probability using the second three bits of the ToS byte. Weighted Random Early Detection (WRED) is used to discard packets based on this value

Notice that because the drop preference (the last digit of the DSCP) has only two bits, its maximum value is three (e.g., AF24 is not a valid DSCP). The default DSCP values are:

- Expedited Forwarding (EF)

- Assured Forwarding (AF11 – AF43)

3.7.1.2 Congestion Management

In the absence of network congestion, packets are transmitted from router interfaces as soon as they arrive. When congestion occurs, packets are received at a router interface faster than they can be transmitted. This causes packets to be queued at the interface until they can be transmitted. Congestion management techniques provide the capability to configure the order at which packets will be transmitted from a congested router interface. This capability is provided by dividing the router interface into multiple queues and directing packets belonging to different service classes to different queues. Each queue may then be serviced at a different rate depending on the relative importance of the service class or classes associated with that queue. A variety of congestion management techniques are commonly implemented, each providing a varied degree of control over packet scheduling.

3.7.1.2.1 First In First Out (FIFO)

When a FIFO scheme is implemented on a router interface, packets are stored in a single queue when the interface becomes congested. Stored packets are forwarded in order of arrival when the network is no longer congested. The operation of a FIFO queue is illustrated in Figure 7 below.

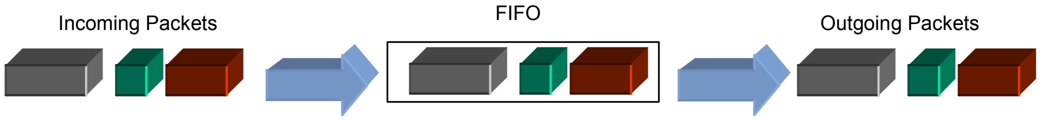


Figure 7
FIFO

The advantage of FIFO is that it's easy to configure. The disadvantages are as follows:

- FIFO utilizes no concept of service classes or priority. All traffic receives the same treatment
- Bursty sources can cause long delays for all traffic received on the interface

3.7.1.2.2 Priority Queuing

Priority Queuing schemes provide for strict enforcement of the packet priorities. The lower priority queues are served only when the higher priority queues are empty. The advantage to using this scheme is that it guarantees adequate bandwidth for real-time applications (i.e., VoIP and VTC). The primary disadvantage of priority queuing is that enforcing strict priority may cause lower priority queues to "starve", depending on the amount of traffic in the higher priority queues. In addition, priority queuing only allows four separate queues to be defined. This limitation restricts the ability to differentiate service between traffic classes.

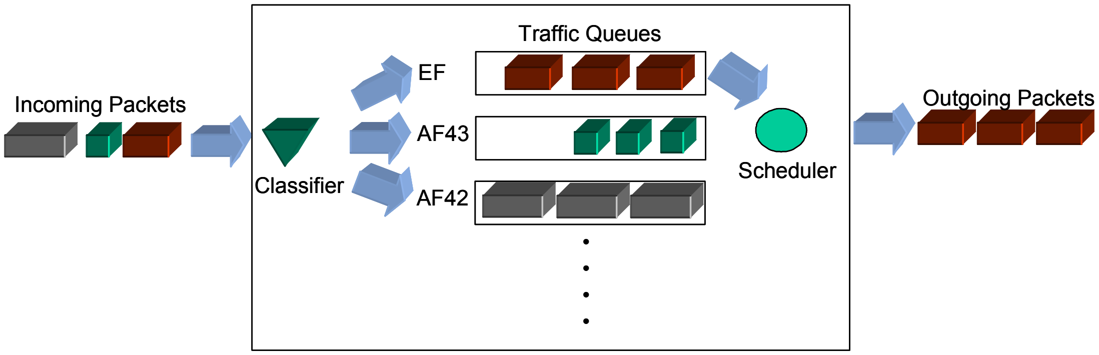


Figure 8
Priority Queuing

3.7.1.2.3 Class Based Weighted Fair Queuing (CBWFQ)

CBWFQ schemes help overcome the limitations associated with priority queuing. CBWFQ implements multiple queues, each with an assigned weight. The weights determine how often the queues are serviced by the scheduler, so some queues can be serviced more frequently than others. The advantages associated with CBWFQ implementations are as follows:

- CBWFQ allows a single priority queue to be defined on an interface to accommodate delay and jitter sensitive traffic such as VoIP. The priority queue may be rate limited to prevent large amounts of traffic in the priority queue from starving the weighted fair queues on the interface
- CBWFQ prevents bandwidth starvation of lower priority queues by allowing a minimum guaranteed bandwidth to be defined through the configuration of queue weights
- CBWFQ allows for the configuration of up to 256 different queues. However, some routers may only support fewer

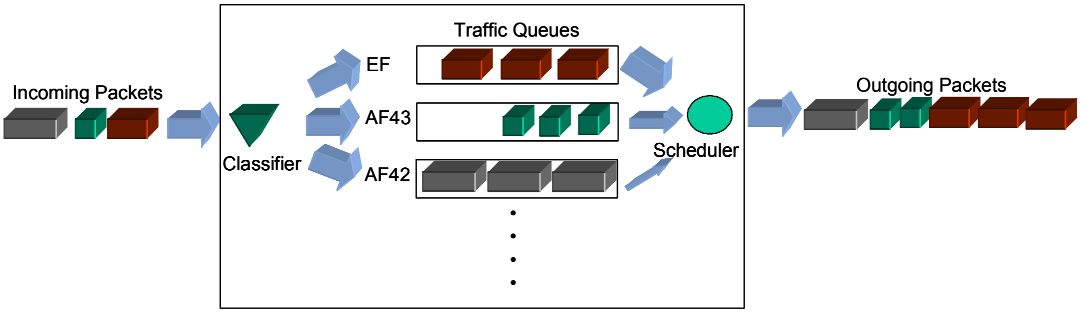


Figure 9
CBWFQ

3.7.1.2.4 Custom Queuing

The Custom Queuing scheme (see Figure 10) assigns a specific amount of memory to each queue and services the queues sequentially. The advantages to this method is it guarantees a user prescribed amount of bandwidth to each queue and it can have up to 16 different queues. The disadvantage is it can produce unpredictable queuing delays.

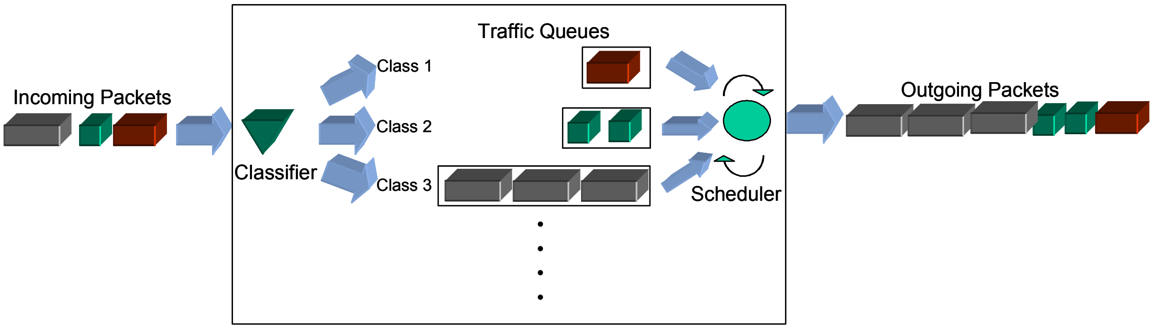


Figure 10
Custom Queuing

3.7.1.2.5 Round Robin Queuing

Like the Custom Queue, in the Round Robin queuing schemes (see Figure 11) the scheduler services the queues sequentially. The different implementations vary in how they de-queue. The advantage to these methods is that they prevent bandwidth starvation of the lower priority queues. The disadvantage is they can produce unpredictable queuing delays.

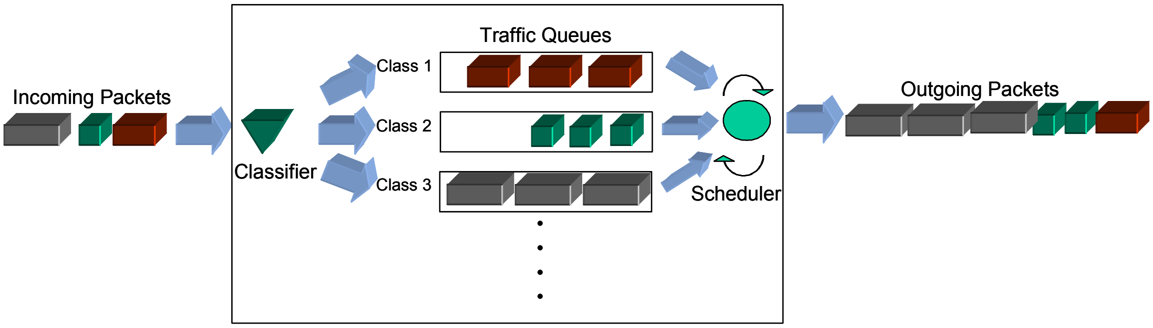


Figure 11
Round Robin Queuing

3.7.1.3 Congestion Avoidance

Unlike the different queuing schemes presented earlier, which are congestion management mechanisms, this section introduces congestion avoidance mechanisms. Congestion avoidance mechanisms seek to proactively avoid congestion at common network bottlenecks by dropping packets. Packet drops occur either randomly or selectively depending on which mechanism is being employed. Generally, the mechanisms presented in this section should only be used with traffic that utilizes transmission control protocol (TCP) at the transport layer. As packets from TCP data flows are dropped by the congestion avoidance mechanism, TCP flow control causes the effected flows to be retransmitted at a slower rate thereby reducing the likelihood that network congestion will occur. Congestion avoidance also improves network efficiency by helping to avoid tail drop. Tail drop occurs when the buffers on a router interface fills to capacity. When the buffer is full, all incoming packets are discarded until the congestion is eliminated. This burst of packet loss engages TCP flow control for all effected flows simultaneously which leads to underutilized links as traffic transmission rates are reduced. The effected flows then slowly increase their transmission rates until congestion occurs once again and the cycle is repeated.

3.7.1.3.1 Random Early Detection (RED)

When RED is enabled for a router queue, it constantly monitors the queue depth using a moving average calculation. Queue depth refers to the number of packets in the queue at any given time. When the average queue depth reaches a certain point, defined by a minimum threshold value, random packet discards begin to occur. The rate at which packets are discarded increases linearly as the queue depth increases beyond the configured minimum threshold value. Once the average queue depth reaches a configured maximum threshold value, the packet discard rate is equal to the mark probability denominator (MPD). The MPD is a configurable parameter that defines the percentage of packets to be dropped when queue depth reaches the maximum threshold. If the MPD is set to 10, 1 out of 10 packets will be dropped at this point. When queue depth exceeds the maximum threshold, all incoming packets are discarded until the queue depth decreases.

3.7.1.3.2 Weighted Random Early Detection (WRED)

The operation of WRED is similar to that of RED except that is uses the drop probability of the DSCP to select which packets to discard; packets with a higher drop probability will be statistically dropped more often than packets with a lower drop probability. WRED allows different queue depth thresholds and MPD values to be set for packets belonging to different service classes.

3.7.1.3.3 Explicit Congestion Notification (ECN)

When ECN is enabled (with either RED or WRED) packets are not dropped. Instead, the last two bits of the TOS byte are be set to “11” for a given packet indicating that the packet experienced congestion somewhere on the network. This in turn, will also cause the TCP flow control mechanism to engage.

3.7.1.4 Policing

Policing is implemented to limit the rate at which certain types of traffic may propagate across a network. Committed access rate (CAR) is commonly used as the rate-limiting feature of the policer. CAR provides control over the maximum rate of traffic that can be sent or received on an interface. CAR typically examines traffic arriving on an interface and matches packets to one or more configured rate policies using DSCP value or other ToS byte marking. CAR policies are implemented using several different rate limit parameters that define the rate at which traffic is allowed to propagate. The three rate limit parameters are described below:

- Average Rate: The allowed long term transmission rate is defined by the average rate parameter. Traffic transmitted below this rate always conforms to the rate limit imposed by the policy
- Normal Burst Size: The normal burst size parameter defines how large a burst of transmitted traffic may be before some of the traffic exceeds the rate limit set forth in the policy
- Excess Burst Size: The excess burst size defines how large traffic bursts may be before all traffic exceeds the rate limit

Once traffic has been examined with respect to the rate limit parameters described above, packets are acted upon by the policer based on the degree to which they conform to the rate limits set forth in the policy. Conforming packets are typically transmitted with no modifications. Packets that exceed or violate the policy may either be dropped or transmitted with modifications. Common packet modifications include the alteration of the ToS byte to cause the packet to receive less favorable treatment by other QoS features implemented at another point within the network.

Policing is one mechanism used to control the rate of traffic entering or a exiting a network segment. Another mechanism that is commonly implemented to control traffic rates is traffic shaping. Similar to policing, traffic shaping utilizes rate limit parameters to control the rate of traffic flows. The primary difference between the two mechanisms is the way in which each deals with the traffic bursts. Policing is able to transmit traffic bursts. In contrast, traffic shaping smoothes out traffic bursts by queuing packets when the traffic rate rises above a certain point. As the transmission rate of the traffic slows, the queued packets are sent from the interface.

3.7.1.5 Signaling

Signaling is used within the Intserv model as a mechanism for a traffic source to inform the network of its need to transmit and negotiate a level of service for the transmission. The most common signaling mechanism used is Resource Reservation Protocol (RSVP**)**.

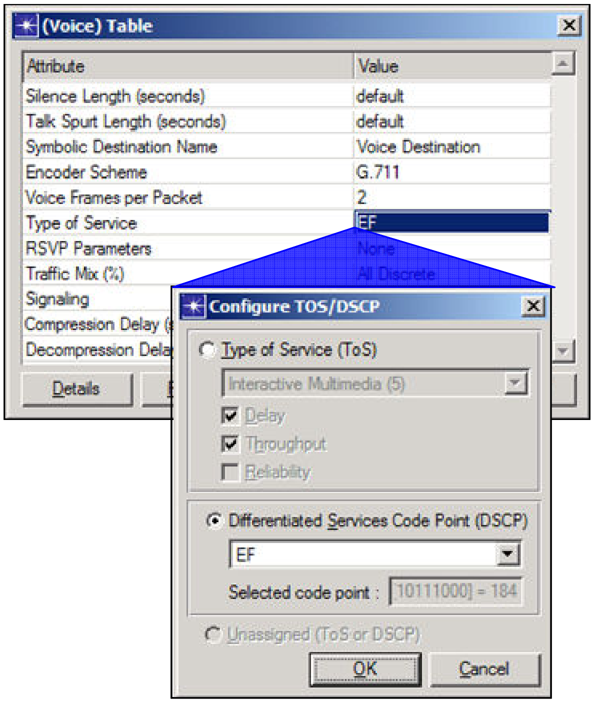
3.7.2 Implementing QoS in JCSS

Implementing QoS in JCSS consist of the following three steps:

1. Classify network traffic
2. Configure IP QoS profiles
3. Apply configured QoS profiles to the network topology

3.7.2.1 Configuring Packet Classification

The majority of the QoS features available in JCSS utilize packet classification to determine what type of treatment to give to a particular packet. Packet classification is generally configured in JCSS by configuring the Type of Service (ToS) parameter located within the traffic model parameters. The contents of the ToS parameter are shown if Figure 12 below.



Attribute	Value
Silence Length (seconds)	default
Talk Spurt Length (seconds)	default
Symbolic Destination Name	Voice Destination
Encoder Scheme	G.711
Voice Frames per Packet	2
Type of Service	EF
RSVP Parameters	None
Traffic Mix (%)	All Discrete
Signaling	
Compression Delay (	
Decompression Dela	

○ Type of Service (ToS)

Interactive Multimedia (5)

☑ Delay

☑ Throughput

□ Reliability

● Differentiated Services Code Point (DSCP)

EF

Selected code point : 10111000] = 184

○ Unassigned (ToS or DSCP)

Figure 12
ToS Parameter

As Figure 12 illustrates, the ToS parameter provides the user with the ability to classify packets using ToS or DSCP values. This method of packet classification configuration is available for the following types of traffic mechanisms:

- Standard Application Models: Standard application models may be accessed and configured through the application utility node. Each application model in the standard library (Email, Voice, Video, etc.), as well as custom application types (e.g. ACE Whiteboard), contains the ToS parameter for defining packet classification
- Application Demands: Application Demands do not represent any particular type of application and provide a quicker way to generate application traffic load in JCSS. Application demands are typically deployed from the application object palette by dragging them between two end station devices. Once deployed in the scenario, an application demand may be right clicked to access the parameters for configuration. Although the parameters available for configuration are more generic than those available for the standard application models, the ToS parameter is identical in structure and function

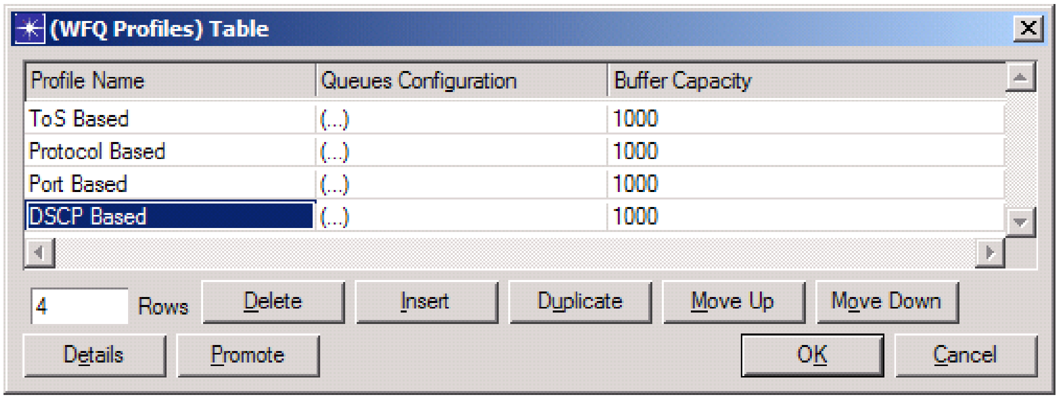
- IP Flows: IP Flows may be deployed and configured using the same procedures described above for application demands. Flows are located within the internet toolbox object palette

3.7.2.2 Configuring IP QoS Schemes

IP QoS schemes may be defined in one of two ways within a JCSS scenario; locally or globally. Local configuration involves the modification of IP QoS parameters within the router devices that will employ QoS. Local QoS configurations are only applicable to the device on which they are defined. QoS schemes may also be configured globally using the QoS configuration utility located within the Configuration OPFAC. QoS profiles defined within the configuration utility object may be applied to any IP interface in a JCSS scenario. Defining QoS globally is generally more efficient than making global configurations, because a global profile may be defined once and applied to many different interfaces. Local configuration requires that each QoS router be configured independently. For this reason, the remainder of this section will focus on the workflow for configuring QoS globally. The next section will provide basic configuration guidance for CBWFQ which is one of the most commonly used congestion management schemes. Many of the parameters described in the following section for CBWFQ are common to the other congestion management schemes as well. Therefore, much of the configuration guidance provided for CBWFQ may be leveraged to configure congestion management schemes in general.

3.7.2.3 Configuring CBWFQ

A variety of CBWFQ profiles are preconfigured within the configuration utility and may be applied to a JCSS scenario without any additional configuration by the user. The default profiles are located within the WFQ Profiles compound parameter and are shown below in Figure 13.

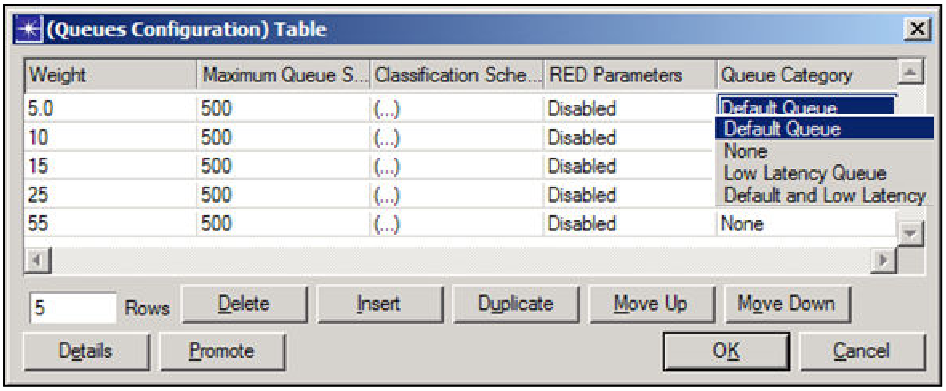


Profile Name	Queues Configuration	Buffer Capacity
ToS Based	(...)	1000
Protocol Based	(...)	1000
Port Based	(...)	1000
DSCP Based	(...)	1000

4 Rows

Figure 13
Default CBWFQ Profiles

In addition to the CBWFQ profiles already available by default, the user may define a custom CBWFQ profile by adding a line to the WFQ Profiles table and configuring its internal parameters. The function of each CBWFQ parameter is described below. For reference, the contents of the default DSCP based CBWFQ profile is shown below in Figure 14.



(Queues Configuration) Table

Weight	Maximum Queue S...	Classification Sche...	RED Parameters	Queue Category
5.0	500	(...)	Disabled	Default Queue
10	500	(...)	Disabled	
15	500	(...)	Disabled	
25	500	(...)	Disabled	
55	500	(...)	Disabled	None

5 Rows

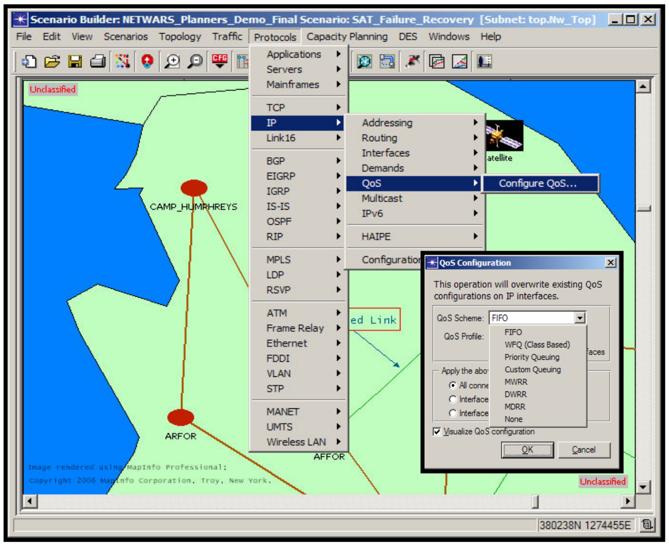
Figure 14
CBWFQ Parameters

The following parameters define the way that each queue will perform within a CBWFQ profile.

- Weight: This parameter defines the rate at which each queue will be serviced during periods of congestion. The relative weight assigned to each queue is proportional to the percentage of the connected link bandwidth that will be allocated to the queue
- Maximum Queue Size: This parameter defines the maximum number of packets that may be stored in each queue during congestion. Once the maximum number of packets has been queued, additional incoming packets will be discarded until congestion is reduced. The sum of the maximum queue size for each queue should generally be equivalent to the overall buffer capacity set for the entire profile
- Classification Scheme: Classification scheme defines the match criteria used to associate incoming packets with each queue. Available choices include DSCP, ToS, IP address, and protocol
- RED Parameters: RED parameters allow the user to define whether or not each queue will implement RED or WRED. Within the parameter, the user may define the thresholds and MPD used define how RED or WRED will behave. It is important to note that each queue may use either RED or WRED, but not both. If neither is defined, the queue will use Tail Drop
- Queue Category: Queue Category allows two special types of queues to be defined with the CBWFQ scheme. Defining a queue as Low Latency Queue (LLQ) causes that queue to enforce strict priority. The LLQ will be serviced at the expense of all other queues and cannot be rate limited. For this reason, the user should be conscious of how much traffic load is assigned to the LLQ. Only one queue may be designated as a LLQ. Additionally, one queue must be designated as the Default Queue. The Default Queue is designed to process any incoming packets that do not meet any of the match criteria for any other queues

3.7.2.4 Applying QoS to Network Interfaces

JCSS provides a configuration wizard for applying global congestion management schemes to interfaces within the network model. Select Protocols > IP > QoS > Configure QoS from the menu, as shown below, to initiate the configuration wizard.



QoS Configuration

This operation will overwrite existing QoS configurations on IP interfaces.

QoS Scheme: FIFO

QoS Profile:

Apply the abo

● All conne

○ Interface

○ Interface

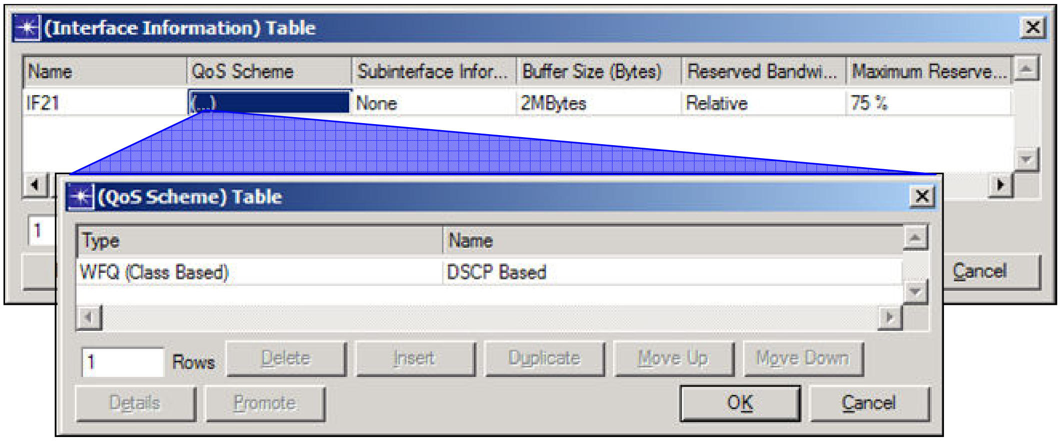
☑ Visualize QoS configuration

Figure 15
QoS Configuration Wizard

As Figure 15 shows, the configuration wizard contains a dropdown menu that allows the user to select which type of QoS scheme to apply to the network. A second dropdown allows the user to select a particular configured profile for each type of scheme. Any profile that has been globally defined in the QoS configuration utility will be available for selection using the wizard. The wizard provides the user with three different options for applying the selected QoS profile to the network. These options are described below.

- All connected interfaces: Selecting this option will apply the selected QoS profile to every connected router interface in the network model
- Interfaces across selected links: Selecting this option will apply QoS to each interface associated with the links selected by the user
- Interfaces on selected routers: Selecting this option will apply QoS to each interface on each router selected by the user

The configuration wizard applies the selected QoS scheme to the network model by configuring the Interface Information parameter of each router selected by the user. The Interface Information parameter is located under IP > QoS Parameters on each router device in the network. A sample configuration is shown below in Figure 16 for the Interface Information parameter.



(Interface Information) Table

Name	QoS Scheme	Subinterface Infor...	Buffer Size (Bytes)	Reserved Bandwi...	Maximum Reserve...
IF21	(...)	None	2MBytes	Relative	75 %

(QoS Scheme) Table

Type	Name
WFQ (Class Based)	DSCP Based

Figure 16
QoS Interface Parameters

In the example shown above, the default DSCP based CBWFQ profile has been applied to interface 21 on the target router. When the profile is applied, a variety of parameters on the interface are configured with default values. The key parameters and their default settings are described below.

- QoS Scheme: This parameter contains the QoS scheme that was selected via the configuration wizard and applied to the network
- Buffer Size: This parameter defines the capacity of the interface buffer in bytes. The default size is 2 MB which is relatively large. The user should be conscious of this value and how it relates to the buffer capacity defined in packets within the QoS profile. If the capacity defined in packets is desired to be the limiting factor, setting the capacity on the interface to a high value is an acceptable strategy
- Max Reserve Bandwidth: This parameter prevents the interface from overflowing at the physical layer by limiting the bandwidth available at the IP layer. If the value is set to close to 100% of the total bandwidth, the physical layer will be overflowed with the addition of layer 2 protocol headers. The default value is 75%. The user should become familiar with this parameter and understand its impact on interface behavior

3.7.3 Additional Lessons Learned

This section provides additional lessons learned and best practices for IP QoS configuration that have not been captured in the previous sections. Some of the guidance provided in this section refers to differences in functionality between global and local QoS configurations. Others refer to QoS features that are unsupported or partially supported. These are based on past experiences using the IP QoS model and do not represent a complete list of all configuration issues associated with the IP QoS model library.

3.7.3.1 QoS Functionality: Global Versus Local

The previous sections of this document have focused on the workflow used to configure IP QoS globally. The JCSS user should be aware that some differences may exist between global and local QoS parameters. In some cases, functionality varies significantly between global and local configurations. Key differences between global and local QoS configurations include the following:

- LLQ: When defined globally, the LLQ within a CBWFQ profile may not be rate limited. This may lead to the starvation of the other queues within the profile if the volume of LLQ traffic is not controlled. In contrast, the LLQ may be rate limited when defined locally on a router. Traffic exceeding the rate limit will not receive priority treatment. This helps protect the other queues in the profile from being starved
- RED and WRED: When defined globally as part of a congestion management profile, RED and WRED profiles must be associated with a particular queue. This removes some flexibility for the user as each service class in a particular queue is impacted by the same RED or WRED settings. In contrast, RED or WRED profiles may be defined for a single service class when defined locally. This allows the user to define unique drop treatments for each class, even for multiple classes within a single queue.
- Match Criteria: The match criteria available to associate packets with queues vary between global and local parameters. One example of this mismatch is IP Address. Global congestion management profiles allow packets to be classified and assigned to queues based on source or destination IP address. This capability is not provided by local congestion management profiles.

3.7.3.2 QoS Interface: Global Versus Local

Each QoS interface in a JCSS scenario may contain global QoS profiles or local QoS profiles. However, global and local profiles do not operate together on the same device interface. For example, if a global CBWFQ profile and a local WRED profile are applied to the same router interface, the local WRED profile will be ignored. Given the differences in QoS functionality provided by global and local configurations, the user should make sure to understand which configuration method best meets the objective of the study.

3.7.3.3 Unsupported Features

Several QoS features are either partially supported or unsupported in the JCSS library. In some cases, the parameters are visible and appear to be available for configuration but are described as unsupported in the parameter details. Unsupported and partially supported features include the following:

- Traffic Shaping: Traffic shaping is not supported in JCSS. The parameters are visible in the local IP parameters of IP devices, but they are not functional. One potential work around involves the configuration of a FIFO queue. By restricting the available bandwidth of an IP interface and configuring the capacity of the FIFO queue, the functionality of Traffic Shaping may be approximated
- Traffic Policing: Most of the policing functionality is implemented. One exception is the Violate Condition. Only the Conform and Exceed Conditions are currently supported in JCSS. Violate is not supported.

3.8 CAPABILITY SCENARIOS

JCSS is shipped with sample *Capability Scenarios* that provides basic implementation of the user community suggested topology configurations developed by the JCSS engineers. These scenarios provide guidance on how different technologies would be implemented within the JCSS environment. The pre-packaged projects include the following:

- JCSS Capacity Planning
- JCSS ETE SATCOM CRYPTO
- JCSS IPv6
- JCSS Link 16 (Contributed Model)
- JCSS MANET
- JCSS Tactical Communications
- JCSS JCAS
- JCSS QoS
- JCSS EPLRS_RS (New JCSS EPLRS Model)
- JCSS JNN

3.8.1 JCSS Capacity Planning

This scenario depicts a traditional Joint Task Force (JTF) and its main components: ARFOR, AFFOR, MARFOR, NAVFOR, and JSOTF. This JTF has reach back capability via Camp Roberts and Ft. Detrick. A NCTAMSPAC site located in Hawaii provides the reach back capability for the NAVFOR component of the JTF.

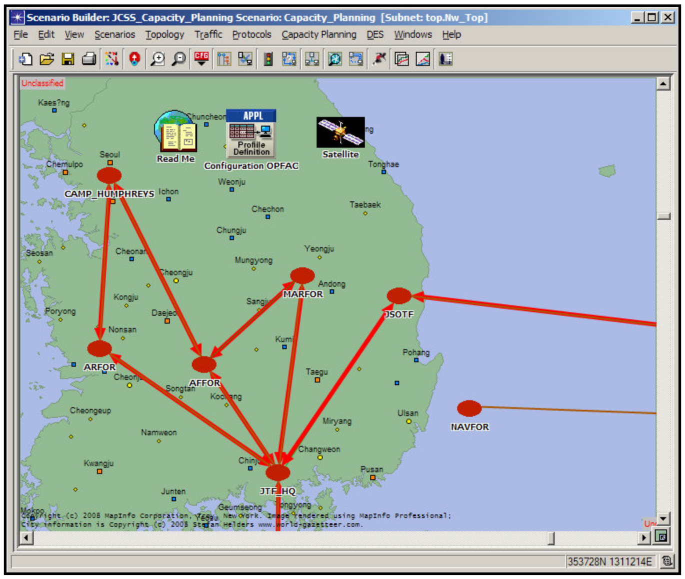


Figure 17
JCSS_Capacity_Planning

As shown in the network diagram each force component has a direct connection to the JTF headquarters except for the NAVFOR, which must communicate via the NCTAMSPAC site.

Each component contains three separate networks: Voice, Data, and VTC. The voice network consists of a ttc-39 connected to a 100-user voice terminal. The data network consists of a Cisco router connected to an Ethernet switch connected to a 100 BaseT LAN. The VTC network consists of an ISDN VTC Terminal connected to an ISDN MCU. All of these networks tie into a multiplexer device, either a Promina or a FCC-100. From the multiplexer the data is then encrypted via a KIV-19 and sent through a wireless link, either satellite or radio. Through these networks, all components in the scenario have the ability to speak to any other component.

3.8.1.1 Traffic

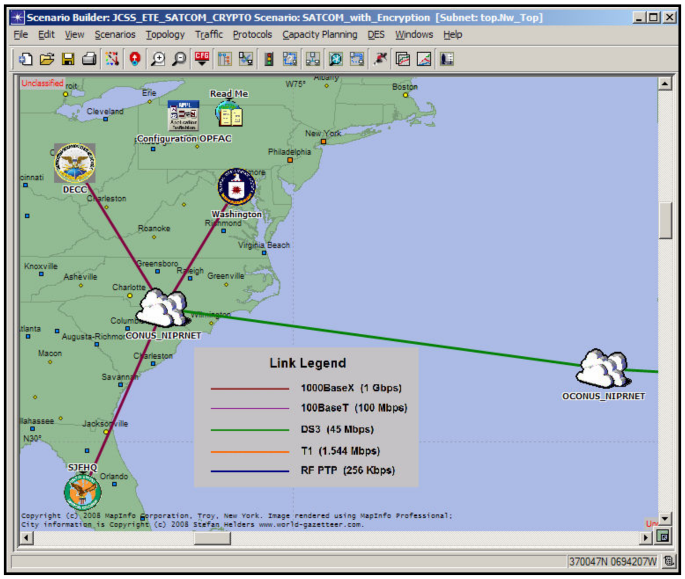
Currently loaded on this network are four types of application traffic: E-mail, HTTP, FTP and Video. In this scenario the E-mail, HTTP, and FTP servers are said to be located at the JTF and the clients are sitting at the each component site.

3.8.1.2 Capacity Planning

After running a capacity planning simulation for a duration of one hour notice that the link between the JTF_HQ and JSOTF is the color red. This is indicating that the link is over 75% utilized in this direction. The JSOTF may require additional bandwidth or a better link than a Microwave LOS connection.

3.8.2 JCSS ETE SATCOM CRYPTO

This scenario depicts an End to End Satellite Communications network architecture containing multiple client sites in various operational environments (e.g. CONUS, OCONUS, and Tactical). Although the network topology contains a number of device models from the OPNET COTS Standard Libraries, it also contains a sampling of device models from the JCSS Standard Libraries (e.g. SATCOM, Encryptors, and Tactical Radios). The NIPRNET backbone is modeled using two IP Cloud devices which abstract the backbone down to three metrics: link bandwidth, packet latency, and packet loss %. The network topology in this scenario is not meant to be a high fidelity representation of any particular real world network or network segment. Rather, it is designed to demonstrate how various device models can be deployed and configured to support a modeling effort.



Scenario Builder: JCSS_ETE_SATCOM_CRYPTO Scenario: SATCOM_with_Encryption [Subnet: top.Nw_Top]

Copyright (c) 2008 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2008 Stefan Helders www.world-gazetteer.com.

Figure 18
JCSS_ETE_SATCOM_CRYPTO

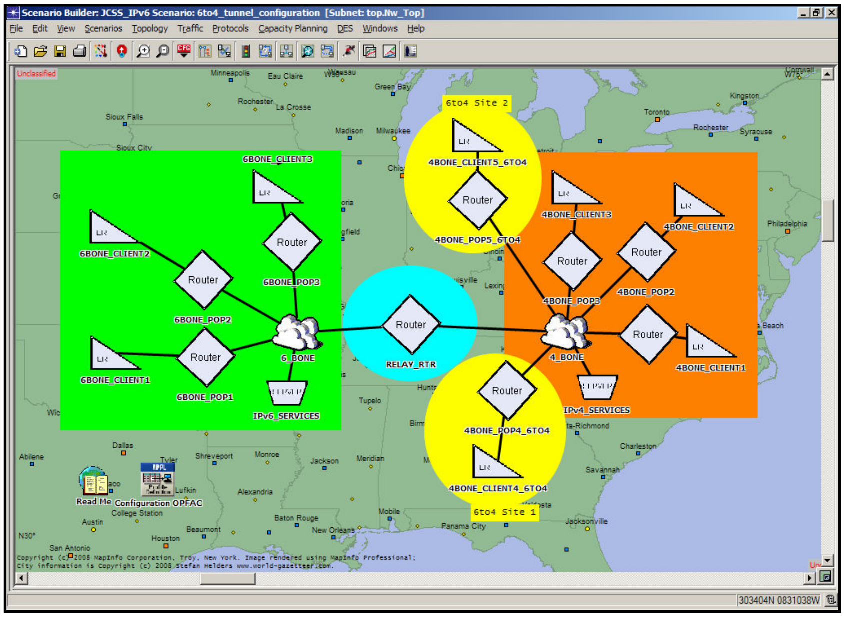
The traffic in this scenario was defined using OPNET COTS Application Models (Standard and Custom). Email, FTP, and ACE Whiteboard traffic has been deployed between each client in the scenario and the enterprise servers located in Columbus, Ohio. In addition, VoIP traffic has been deployed between three different conversation pairs: CONUS to CONUS, CONUS to OCONUS, and CONUS to Deployed JTF. Result panels have been provided that demonstrate the differences in application performance as a function of client location within the network topology.

Additional information about the specific device model configurations used in this scenario is available inside of various OPFACs in the scenario. Please refer to the following locations to access this information:

Model Type	Location of Model Information
Client	SJFHQ OPFAC: Tampa, FL
Server	DECC OPFAC: Columbus, OH
Encryption	STEP_SITE OPFAC: Bahrain
SATCOM	STEP_SITE OPFAC: Bahrain
EPLRS	JFSOCC OPFAC: Iraq

3.8.3 JCSS IPv6

The objective of this Project is to demonstrate the configuration necessary to integrate IPv4 and IPv6 networks within the JCSS tool. This was accomplished using a combination of IPv6 to IPv4 tunneling and static routes.



Scenario Builder: JCSS_IPv6 Scenario: 6to4_tunnel_configuration [Subnet: top.Nw_Top]

Figure 19
JCSS_IPv6

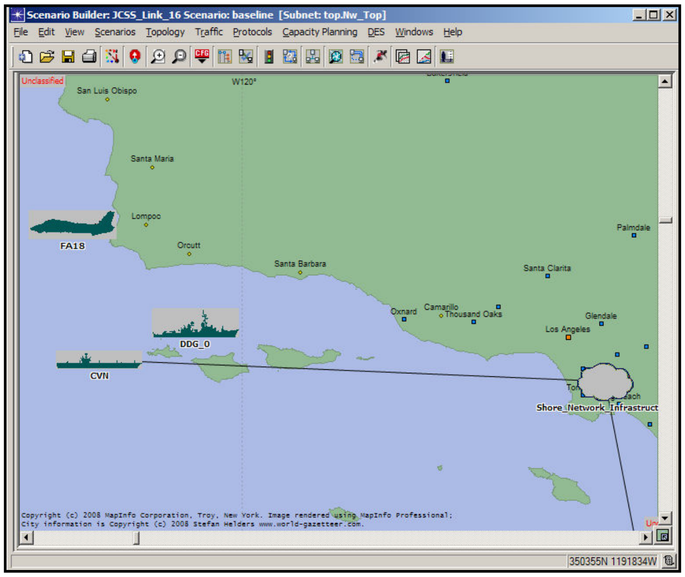
The network topology used in this scenario includes the Native IPv4 and IPv6 Architectures described in other scenarios of this project. In addition, a relay router and two 6to4 IP sites have been added. The relay router and 6to4 POP routers contain special configurations that allow IPv6 traffic to be tunneled through the IPv4 backbone. Static routes have been configured to allow communication between 6to4 clients and IPv6 clients. More detailed IP configuration information is located in the following OPFACs: 4BONE_CLIENT4_6TO4, RELAY_RTR, and 4BONE_POP4_6TO4.

Through the IP configurations described above, the 6to4 clients in this scenario are able to communicate over the 4BONE to other 6to4 clients, IPv4 clients, and IPv6 clients. The result panels in this scenario show the throughput generated by VoIP traffic for each of those cases. The throughput varies depending on both the client pairs that are communicating and the link that the throughput statistic is collected on. The variation in throughput is caused by the difference between Pv4, IPv6, and 6to4 encapsulation.

An IPv6 license must be available in order to execute DES with this scenario and any other scenario that contains IPv6 addressing. For more information about the OPNET IPv6 model, consult the IPv6 User Guide available through the Help Menu in the JCSS GUI.

3.8.4 JCSS Link 16

The Network Centric Warfare Analysis branch of the Space and Naval Warfare (SPAWAR) Systems Center, San Diego (SSCSD) is the lead Navy model development group for the Network Warfare Simulation (NETWARS) tool. NETWARS has been designated by the Navy Modeling and Simulation Management Office (NAVMSMO) as the primary tool for communications Modeling and Simulation (M&S). One of the products resulting from the Navy's use of NETWARS is a Link 16 model. The NETWARS Link-16 Model User Guide provides guidance to the user for configuration of this Link 16 model suite within NETWARS simulations.



Copyright (c) 2008 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional; City information is Copyright (c) 2008 Stefan Helders www.world-gazetteer.com.

Figure 20
JCSS_Link_16

3.8.5 JCSS MANET

The objective of this project is to demonstrate the ability to use OPNET Wireless LAN (WLAN) and Mobile Ad Hoc Networking (MANET) models in JCSS to model IP enabled tactical networks. Each scenario in this project contains the same network topology. The topology is comprised of two squads of infantry, two infantry carrier vehicles (ICVs), an unmanned aerial vehicle (UAV), and a mobile service delivery node (SDN). Each of the two squads are represented by a cluster of nine soldiers (OPFACs) that attempt to pass traffic across the topology to the SDN through a series of intermediate hops. Those intermediate hops are provided by the ICVs and the UAV. Each of the nodes in this scenario is represented by a WLAN device model located in the WLAN palette. The devices support the use of 802.11 wireless Ethernet Standards at layers 1 & 2 and various Ad-Hoc Routing Protocols at layer 3.

While each scenario contains the same network topology, two different ad hoc routing protocols and two different degrees of node mobility are utilized in this project. The project is designed to examine the performance implications of using Ad Hoc On Demand Distance Vector (AODV) versus Optimized Link State Routing (OLSR) ad hoc routing protocols across different mobility cases. The mobility cases will be described in greater detail within each individual scenario. In order to view the OPFACs traversing their configured trajectories, select View > Show Time Controller from the Scenario Builder menu and use the "VCR" buttons to control the motion display.

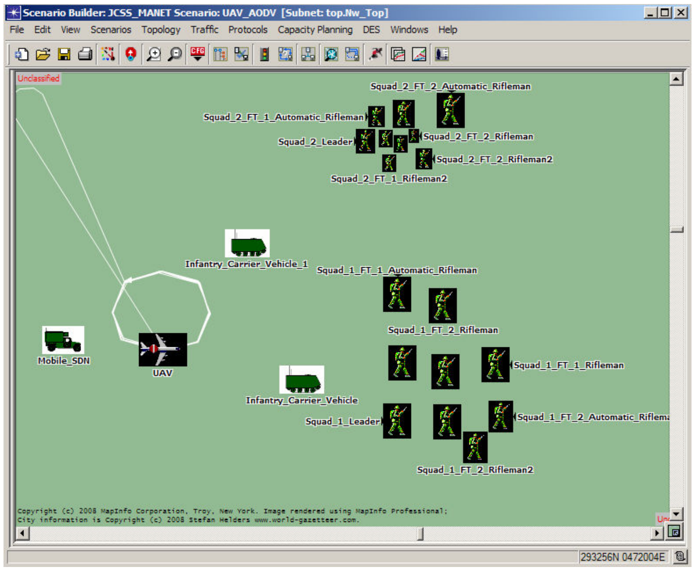

Scenario Builder: JCSS_MANET Scenario: UAV_AODV [Subnet: top.Nw_Top]
File
Edit
View
Scenarios
Topology
Traffic
Protocols
Capacity Planning
DES
Windows
Help
Unclassified
Squad_2_FT_2_Automatic_Rifleman
Squad_2_FT_1_Automatic_Rifleman
Squad_2_Leader
Squad_2_FT_2_Rifleman
Squad_2_FT_2_Rifleman2
Squad_2_FT_1_Rifleman2
Infantry_Carrier_Vehicle_1
Squad_1_FT_1_Automatic_Rifleman
Squad_1_FT_2_Rifleman
Mobile_SDN
UAV
Squad_1_FT_1_Rifleman
Infantry_Carrier_Vehicle
Squad_1_Leader
Squad_1_FT_2_Automatic_Riflema
Squad_1_FT_2_Rifleman2
Copyright (c) 2008 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2008 Stefan Helders www.world-gazetteer.com.
293256N 0472004E

Figure 21
JCSS_MANET

An OPNET Wireless License is required in order to execute DES for this scenario or any other that utilizes the WLAN model library.

The nodes in this scenario utilize AODV as the ad hoc routing protocol. The only node in motion is the UAV which circles for a duration of time to provide an intermediate hop between the two infantry squads and the SDN. About 27 minutes into the simulation, the UAV moves away from the rest of the network and communication between the squads and the SDN is lost until the UAV returns to its previous flight pattern approximately 20 minutes later.

Additional information regarding the configuration of this scenario is located inside of the SDN OPFAC. Information about the DES Result Panels in this project is located inside of the DES_Results object.

3.8.6 JCSS Tactical Communications

This scenario depicts a hypothetical tactical scenario in which three humvees are circling an area and periodically transmitting voice traffic to one another via SINCGARS radio devices. Each humvee in the scenario contains a SINCGARS radio. The three humvees communicate with each other through a broadcast network which has been configured with typical parameter settings for SINCGARS communication. The parameter settings of the broadcast network can be viewed and edited by right clicking on the broadcast network object changing the values of the configurable fields. Changing the parameters of the broadcast network will automatically update the corresponding parameters of any connected radio devices.

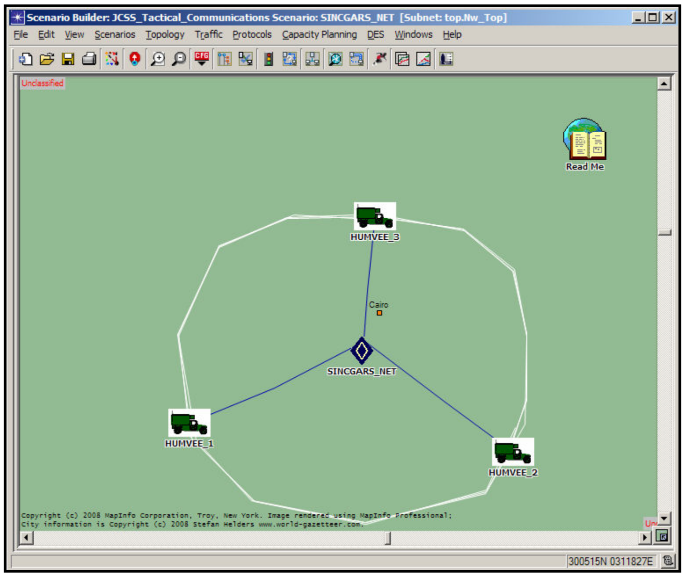

Scenario Builder: JCSS_Tactical_Communications Scenario: SINCGARS_NET [Subnet: top.Nw_Top]
File Edit View Scenarios Topology Traffic Protocols Capacity Planning DES Windows Help
Unclassified
Read Me
HUMVEE_3
Cairo
SINCGARS_NET
HUMVEE_1
HUMVEE_2
Copyright (c) 2008 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2008 Stefan Helders www.world-gazetteer.com.
300515N 0311827E

Figure 22
JCSS_Tactical_Communications

The traffic in this scenario is represented using Information Exchange Requirements which have been applied to each of the three humvee OPFACs in the scenario. In this case the IERs have been configured as Voice which is one of the data types that SINCGARS supports. The IERs can be viewed by right clicking on any of the three OPFACs and selecting View IERs from the menu.

The mobility pattern of each humvee OPFAC in this scenario has been configured using the trajectory GUI. The trajectory GUI can be opened by selecting the Define Trajectory option under the Map Menu in JCSS Scenario Builder. The mobility of the OPFACs in the scenario can be visualized using the Show Time Controller option under the View menu.

3.8.7 JCSS_JCAS

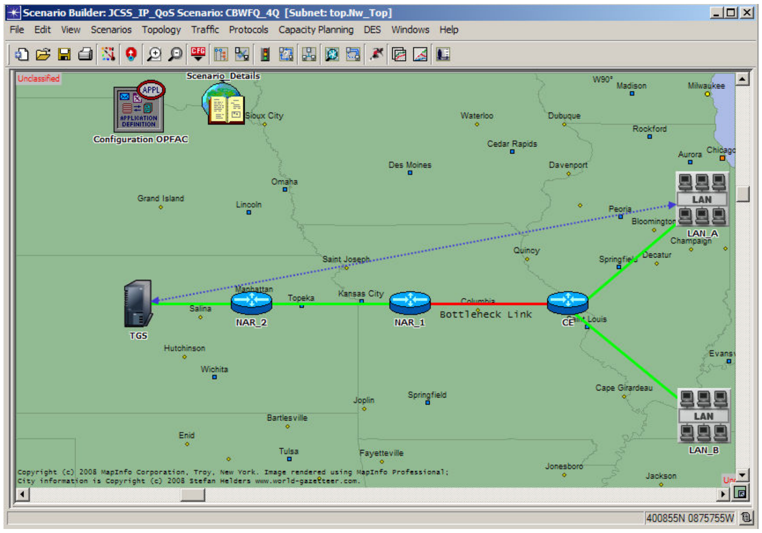
A Joint Close Air Support (JCAS) scenario was developed in NETWARS to represent an operationally relevant scenario at the tactical edge configured with wireless communications assets. The DISA DISN-IF utility was used to create the portion of the network model that represents the DISN capability in CONUS along with the satellite interface assets.

The overall JCAS situation is as follows: A preplanned CAS Target on the Joint Integrated Priority Target List (JIPTL) is on the targeting schedule as a lower priority target for engagement. Dismounted USMC forces on patrol have discovered increased activity at this particular target site and made assessment that the activity was such that it mandated an increase in priority to conduct engagement. The JCAS scenario is comprised of 12 nodes including an F-15E, an F/A-18F, a Joint CW E-3 Sentry and 9 ground based nodes.

Two scenarios are provided, “tactical_edge” and “end_to_end”. The tactical edge scenario has all nodes situated in theatre. In the end_to_end scenario, the USMC_DASC node is moved to CONUS and has connectivity to the theatre via the DISN. For more information on the JCAS scenario, please refer to the JCAS Model User-guide.

3.8.8 JCSS_QoS

This scenario contains a variety of IP QoS functions that are designed to provide differentiated QoS to each of the configured service classes. The QoS functions configured within this scenario are summarized below. Additional configuration details are available at various locations within the network model.



Copyright (c) 2008 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2008 Stefan Helders www.world-gazetteer.com.

Figure 23
JCSS_IP_QoS Scenario

- **Packet Classification & Differentiation (DSCP):** Packets are grouped into different service classes based on their DSCP value and receive different QoS accordingly. The DSCP is configured within the application level attributes for each of the various types of traffic configured in this scenario.

The DSCP values are assigned as follows:

Type	**DSCP**
----------	----------------
VoIP	EF
User Signal	EF
Ntwk Control	192
OAM	64
Video	AF43
HPHTD	AF31
SHTD-PP	AF11, AF12, AF13
SHTD-NW	AF21, AF22, AF23
Best Effort	0

- **IP Policing:** IP Policing is implemented to control the rate at which certain traffic flows may be transmitted. IP Policing is configured locally on the NAR 2 router. Additional details about the configuration of the Policer is provided within the NAR 2 OPFAC.

- **Class Based Weighted Fair Queuing (CBWFQ)**: CBWFQ divides the resources of a router interface into a configurable number of weighted fair queues. Each queue is configured with a weight which provides a bandwidth guarantee on the router interface. CBWFQ is configured locally on the NAR 1 router. In this scenario, the router interface is only divided into 4 queues. The NAR 1 OPFAC contains additional details about the CBWFQ configuration.

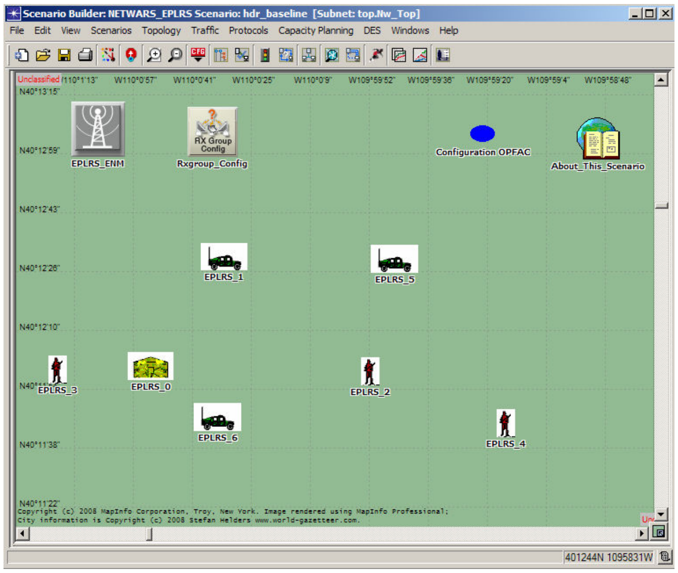
- **Weighted Random Early Detection (WRED):** WRED is implemented for certain queues in the network to avoid globalization problems that occur as router queues overflow. Without WRED, packet drops resulting from queue overflows cause multiple TCP hosts to slow transmission simultaneously. During this time, the link is underutilized. Transmission rates then slowly scale back up until the buffers overflow and the process is repeated. WRED avoids these issues by dropping packets for certain TCP flows once the queue reaches a certain size. This causes the affected hosts to reduce transmission rate and frees up resources of other flows. WRED is implemented on NAR 1 router. Additional details about WRED configuration are provided within the NAR 1 OPFAC.

3.8.9 JCSS_EPLRS

JCSS 7.0 introduces a new high fidelity EPLRS modules that provides support for different types of needlines including HDR and CSMA. The EPLRS ENM (EPLRS Network Manager) and EPLRS RS (Radio System). For additional information, please refer to the EPLRS Model User-guide and sample EPLRS scenario that ships with the software.
In a CSMA Needline, all radio share the same time/frequency resources. Any radio can be a source and is the most widely used needline type. HDR is a predefined one-to-one communication where time and frequency are reserved for each HDR needline.

In this scenario, data IERs are being sent by the EPLRS devices in EPLRS_3, EPLRS_4, EPLRS_5 and EPLRS_6. The data IERs are not based on specific traffic, but rather it is acting as a method to validate connectivity between EPLRS devices in this scenario.
At the starts of the simulation, EPLRS_6 begins to move east in an effort to get out of range on of the transmitter and relays. Around 106 seconds, EPLRS_6 successfully moves out of range, but then EPLRS_1 arrives and acts as a relay, so the route from 3 to is reestablished, allowing for some IERs to arrive perished.

Once EPLRS_6 gets out of range of EPLRS_1, the network has to wait for EPLRS_5 to move into range, even though EPLRS_2 is in range to act as a relay. This is because EPLRS_2 isn't part of the Needline, and this can be seen in the ENM Node's attributes under "Needline Definition". For any given scenario, the EPLRS devices must have a ENM node deployed for them to work properly.



Copyright (c) 2008 MapInfo Corporation, Troy, New York. Image rendered using MapInfo Professional;
City information is Copyright (c) 2008 Stefan Helders www.world-gazetteer.com.

Figure 24
JCSS_EPLRS_RS Scenario

Unlike the CSMA Scenario, since the HDR Needline defines two endpoints as the starting and stopping point of traffic, EPLRS_3 and EPLRS_5 cannot send IERS to each other, causing some IERs to fail. This can also be observed in the ENM node, as the ENM node does not contain a HDR Needline between EPLRS_3 and EPLRS_5. If the ENM node is modified to include a Needline between the two, then the traffic between the two EPLRS devices would not fail.

Expected Scenario Results

```
Scenario IER Summary
--------------------
Total IER Sent: 39
Total IER Received: 31
Total IER Failed: 7
Total IER Undelivered: 1
Total IER Perished: 3
```

Not all IERs sent should be received. Since this is a HDR needline, only endpoints can pass traffic to each other. Even if a radio is a relay member of the specific needline, it will not be able to send or receive traffic, as it will be not listening as a receiver, but acting as a relay. As a result, the IERs between EPLRS_5 and other radios fail, as can be observed in the IER Results.

Three IERs perish. This is because when the IER initially transmits, the destination radio is out of range. As a result, the IER attempts to retry, and when the IER finally succeeds, the time it took for the delivery of the IER is greater than the IER's Perishibility.

3.8.10 JCSS_JNN

In this scenario, both an Air Force base and Deployed Army Batallion communicate to sites in the United States from the Middle East. Flows are deployed as well as applications and IERs to put traffic on this network. For each OPFAC or Organization, a template was used that is available with this release of the software.

For the JNN Hub and JNN Organizations, the JNN Army Batallion templates were used and modified, removing excess models that are not required of the scenario. For the Step and

CONUS, the DoD_Gateway_STEP template was used. For Fort_Belvoir and Langley, the CITS Block 30 templates were used.

Using available templates in the software, it is possible to rapidly deploy set configurations into several different scenarios, saving time spent reconfiguring models when preparing scenarios for traffic analysis, either in Capacity Planner or DES.

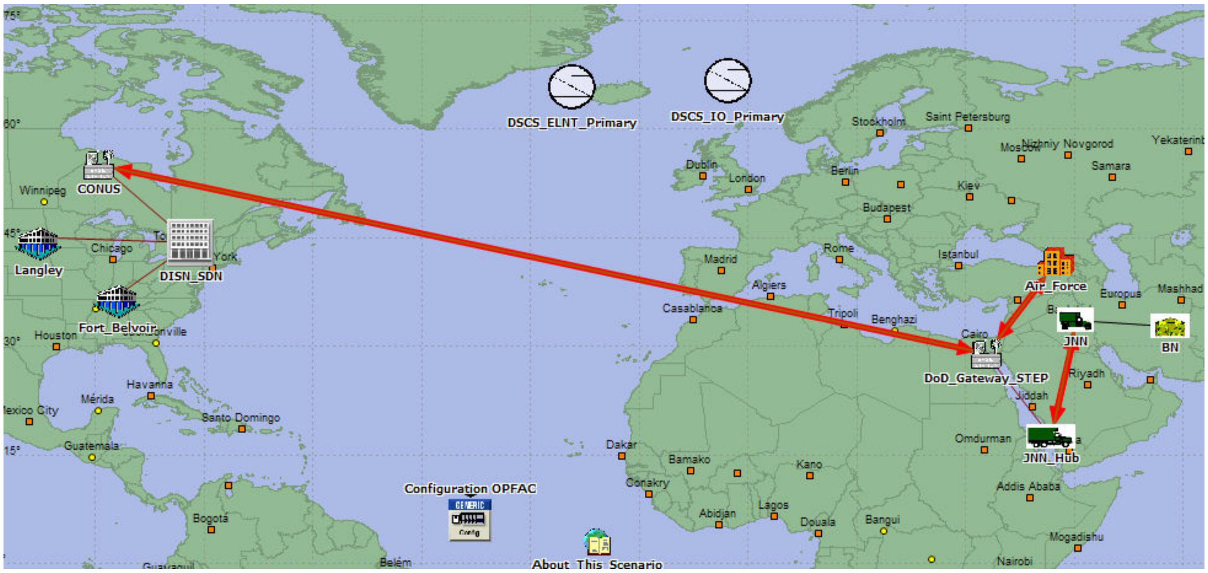


Figure 25
JCSS_JNN Scenario

APPENDIX A – NETWARS SYSTEM OVERVIEW

NETWARS consists of three primary elements: (1) databases or libraries (including Device Models, OPFACs, generic organizations, and IERs), (2) a Scenario Builder graphical user interface (GUI), and (3) a Simulation Domain (which is based on the OPNET commercial Simulation Engine). The relationship of these primary components is shown in the NETWARS Architecture diagram in Figure A-1.

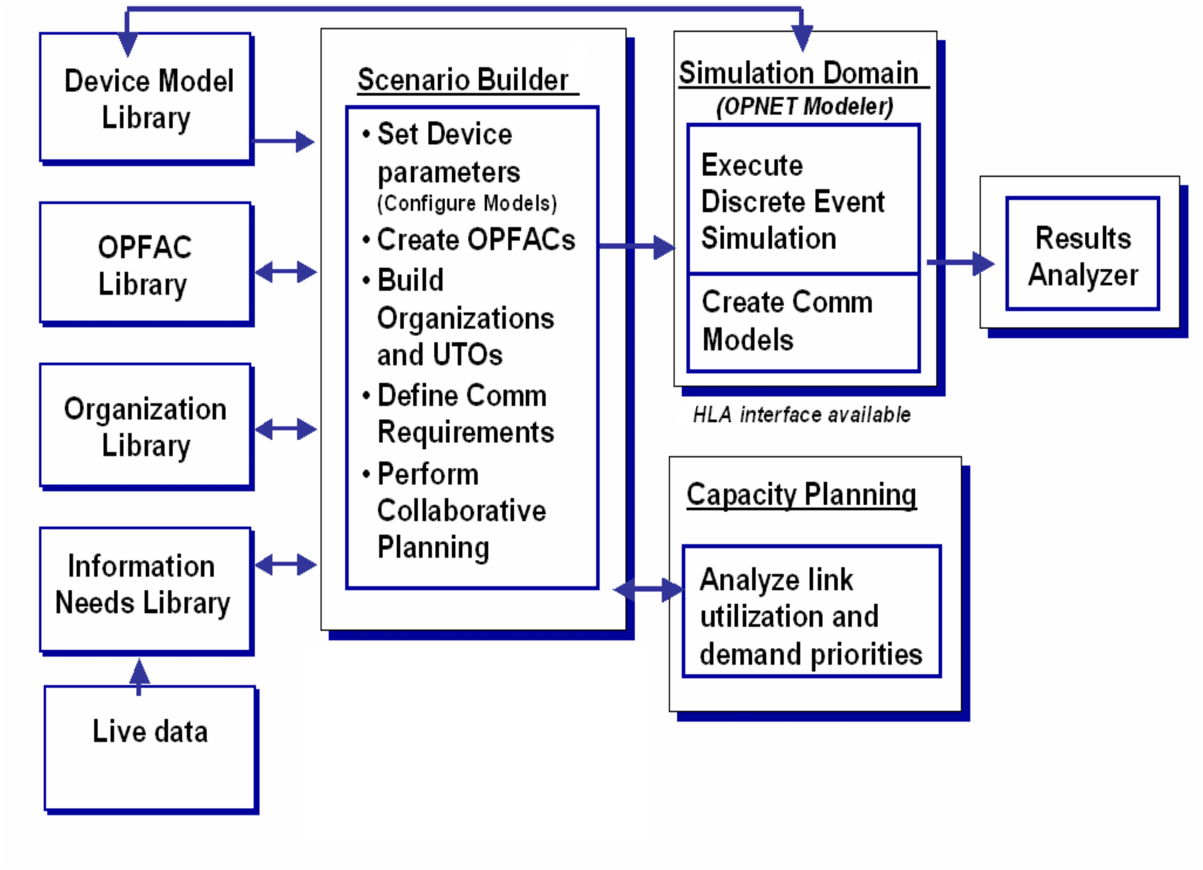

Device Model Library
OPFAC Library
Organization Library
Information Needs Library
Live data
Scenario Builder
• Set Device parameters (Configure Models)
• Create OPFACs
• Build Organizations and UTOs
• Define Comm Requirements
• Perform Collaborative Planning
Simulation Domain
(OPNET Modeler)
Execute Discrete Event Simulation
Create Comm Models
HLA interface available
Results Analyzer
Capacity Planning
Analyze link utilization and demand priorities

Figure A-1 -NETWARS Architecture

3.9 LIBRARIES

Externally provided libraries are essential for the successful formulation and execution of models to answer specific analytical questions. The Scenario Builder uses the Device Models, OPFAC, organizations, and IER libraries to develop detailed descriptions of operational deployments and scenarios.

Device Model Library

CDMs are representations of communications equipment that are used by both the Scenario Builder and the Simulation Engine. CDMs have the logic and attributes of specific COTS and military communications equipment (routing schemes, number of ports, protocols, priorities and, for wireless networking, RF frequency bandwidth, waveform, and Tx/Rx performance). The NETWARS Device Model library consists of COTS CDMs provided with OPNET and custom CDMs developed as part of the NETWARS program. Custom CDMs are either jointly developed or Service-developed and are based on specific study needs. Custom CDMs often represent unique military communication devices that have no commercial equivalent, and therefore they should be developed by the military service that is responsible for the device the model represents. Modelers must use the specific interface standards contained in the *NETWARS Model Development Guide* to develop custom CDMs to satisfy their unique device requirements. Only NETWARS-compliant CDMs can be used to produce and consume NETWARS IERs. In addition, both COTS and custom CDMs must be editable to incorporate attributes assigned by the Scenario Builder.

OPFAC Library

OPFACs are representations of communications equipment, and intra-nodal connections, some of which are capable of movement. Each OPFAC has its own unique traffic flow. Multiple OPFACs can exist within the same vehicle or platform, and each OPFAC can contain one or more CDMs. OPFACs stored in the library database can be used in scenarios either as is or modified within the Scenario Builder. In addition, OPFACs can be created within the Scenario Builder.

Organizations Library

Organizations are groups of OPFACs that are structured and linked by network relationships. Organizations reflect the structures and relationships inherent in current communications doctrine, as it relates to the OPFACs in an organization. However, they may also reflect experimental relationships, which are being studied.

IER Library

IERs represent elemental communications requirements between OPFACs that are based on mission(s) and mission phase(s). The traffic generated from IERs is specific to the scenario being studied and can be either externally generated or reused from the library database. IERs contained in the NETWARS IER library can be explicitly invoked according to OPFAC association or implicitly invoked based on organizational relationships. In addition, study-specific IERs can be imported from external tab delimited text files (e.g., Microsoft Excel spreadsheets) or defined by the user.

3.10 SCENARIO BUILDER

The Scenario Builder allows the warfighter to incorporate deployment scenarios, organizational structures, unit representations, and IERs (provided by the libraries). It provides a GUI and has an editing capability that allows the planner and analyst to execute repeated what-if scenarios. The Scenario Builder provides a means of combining CDMs into an OPFAC and allows intra-nodal connections that facilitate the combining of multiple OPFACs into an organization. A capability to analyze radio line-of-sight connectivity is also included in the Scenario Builder. In addition, the GUI provides the capability of importing organizations and previously created OPFACs and IERs.

The Scenario Builder also allows users to plan for and develop C4I architectures, including transmission systems, technical control facilities, voice, data, video teleconferencing (VTC), and message switch networks. It provides a semi-collaborative planning process by allowing individually developed portions of architecture to be integrated into a larger more complete architecture.

3.11 CAPACITY PLANNER

The Capacity Planner provides planners with the ability to conduct quick-turnaround analyses, without the need of a detailed simulation or a set of OPNET-specific CDMs. In a rapid planning environment, the accuracy (fidelity) afforded by simulation is traded off against the time-value of immediate answers. The results are reflected within the Scenario Builder GUI as a degree of satisfaction of the IERs.

The Capacity Planner uses analytic techniques, as opposed to the discrete event Simulation Engine, to provide support to communications planners who must compare their mission planning documents, including information requirements plans, against the existing operational environment to identify shortfalls in information support.

3.12 SIMULATION DOMAIN

The Simulation Domain consists of the Simulation Engine and an SCM. The SCM allows the Scenario Builder to provide its output to a COTS Simulation Engine (e.g., OPNET Simmulation Runtime Engine) in a file structure expected by the Simulation Engine. The SCM translates organizational representations and information flows into discrete events between sender-and-receiver pairs tied to specific communications equipment representations in the Simulation Engine.

The Simulation Runtime Engine is a COTS product developed by OPNET Technologies, Inc. It takes the scenario representation from the Scenario Builder and environmental factors and then generates and processes events to obtain the needed results. These results are provided to the Results Analyzer for display and interpretation.

3.13 RESULTS

The results of the NETWARS analysis are presented as a series of MOPs. In general, MOPs provide a system's user with information on how well the system performs its functions in a given environment (e.g., number of targets detected, reaction time, and task completion time). The NETWARS MOPs focus on predicting the ability of selected communications equipment to satisfactorily send or receive information (e.g., file transfer, situation awareness update, e-mail message, database transfer), to the extent to which the CDMs utilized from the NETWARS library accurately represents the equipment.

APPENDIX B – SIMULATION LOG ENTRIES

As described in Section 3.4.2.2, the simulation generates a simulation log detailing potential problems or issues. The following section lists some of the more common TCP-, OSPF-, and IP-related entries that a NETWARS user may encounter.

3.14 TCP-RELATED ENTRIES

SYMPTOM:
Unexpected TCP connection-based results for window size related statistics

REASON:
The value assigned to "TCP Receive Buffer Capacity" attribute: N bytes exceeds the maximum allowed value. Note that the maximum allowed value when both ends of the connection enabled the "Window Scaling" option is (65535 x 214) or 1,073,725,440 bytes. Otherwise, it is 65535 bytes. Setting the value of receive buffer to size N bytes.

SUGGESTION:
1. If larger windows are desired, then enable "TCP Window Scaling Option" (if not already on).

2. If "TCP Window Scaling Option" is used, then note that the max possible value is limited to (65535 x 214) or 1,073,725,440 bytes, provided both ends support window scaling.

SYMPTOM:
1. High TCP traffic overhead

2. Acks are not being piggybacked with data

POSSIBLE CAUSES:
"Maximum Ack Delay" model attribute is set to 0.0

SUGGESTIONS:
1. Set "Maximum Ack Delay" to a non-zero value.

2. Set "Delayed ACK Mechanism" to "Segment\Clock Based".

SYMPTOM:
TCP is retransmitting data segments, which will cause additional overhead on the lower layers and links

CAUSE(S):
This may be normal behavior on a network, which has either:

1. Slow links
2. Possibly dropped packets (ATM or frame relay [FR])
3. Queuing delays

SUGGESTIONS:

If any of the causes listed above are expected, no action is required, as this is normal protocol behavior for TCP. Otherwise look for problems in the lower layers, such as IP dropped packets or other errors in the simulation log. View TCP response time statistics for data regarding the delays.

ERROR(S):
Unable to interface TCP with application

POSSIBLE CAUSES:
No stream connection between application and TCP

SUGGESTIONS:
Review node configuration and add streams to connect application module to TCP.

ERROR(S):
No local port specified in TCP OPEN resulting in:

1. TCP connections not being established
2. Data rates or traffic lower than expected

POSSIBLE CAUSES:
Application is not maintaining correct state information about connections

SUGGESTIONS:
Look in Simulation Log for error messages from the application above TCP.

ERROR(S):
Application attempted to open multiple connections with same specifications
Local Port: X
Remote Port: Y
Remote Address: nnn

POSSIBLE CAUSES:

1. There are multiple "passive" connections running for the attempted service type. For example, in cases where TCP connection is aborted in the connection establishment phase (quite common for heavy HTTP loads).
2. Application is not maintaining correct state information about connections

SUGGESTION(S):

1. Allow sufficient time to complete TCP connection setup (e.g., lower the application traffic generation rate). This behavior will be experienced with application like http. For this case, ignore the message, as it is a valid TCP protocol behavior.
2. Look in Simulation Log for error messages from the application above TCP.

ERROR(S):
An OPEN request was received for an existing connection, resulting in:

1. TCP connections are not being established
2. Data rates or traffic is lower than expected Connection ID: nnn

POSSIBLE CAUSES:
Application is not maintaining correct state information about connections

SUGGESTIONS:
Look in Simulation Log for error messages from the application above TCP.

ERROR(S):
A TCP OPEN request was not processed.

POSSIBLE CAUSES:

1. No local port specified in OPEN command
2. Memory shortage on simulation system

SUGGESTIONS:
Look in Simulation Log for error messages generated in TCP at the same simulation time.

ERROR(S):
TCP <cmd> did not specify valid connection ID: nnn

POSSIBLE CAUSES:
Protocol errors in higher layer application

SUGGESTIONS:
Check the Simulation Log for errors in the application.

SYMPTOM(S):
TCP RST received for invalid connection. Packet destroyed. This message will not be repeated.

POSSIBLE CAUSES:
RST was for connection, which had already closed and timed out. This is normal protocol behavior for TCP, following retransmissions and subsequent ACKs being sent.

SUGGESTIONS:
Check the Simulation Log for retransmissions, which indicate possible long delays.

SYMPTOM(S):
TCP ACK received for invalid connection. Packet destroyed and RST sent. This message will not be repeated.

POSSIBLE CAUSES:
ACK was for connection, which had already closed and timed out. This is normal protocol behavior following retransmissions or after higher layer aborts a TCP session.

SUGGESTIONS:
Check the Simulation Log for retransmissions, which indicate possible long delays.

ERROR(S):

SYN (open) received for invalid connection. Responding with RST ACK.
Local Port: X, Remote Port: Y
Remote Address: nnn

POSSIBLE CAUSES:
The connection had not yet been created on this node through active or passive open

SUGGESTIONS:
Verify that this node should be the target of an OPEN request, i.e., a server.

ERROR(S):
Segment received for invalid connection. Responding with RST ACK.
Local Port: X, Remote Port: Y
Remote Address: nnn

POSSIBLE CAUSES:
Segment received after the connection had closed and timed out

SUGGESTIONS:
Verify that this node should be the target of an OPEN request, i.e., a server.

ERROR(S):
Unknown status received for connection ID nnn.

POSSIBLE CAUSES:
Protocol errors in higher layer application

SUGGESTIONS:
Check the Simulation Log for errors in the application.

SYMPTOM(S):
TCP has reached the limit on retransmission attempts. The connection will reset and an abort will be issued to the application.

CAUSE(S):
This may be normal behavior on a network, which has either:

1. Slow links
2. Possibly dropped packets (ATM or FR)
3. Queuing delays
4. Link/Node failures

SUGGESTIONS:
If any of the causes listed above are expected, no action is required, as this is normal protocol behavior for TCP. Otherwise look for problems in the lower layers, such as IP dropped packets or other errors in the simulation log. View TCP response time statistics for data regarding delays. The current setting for the limit on retransmissions is nnn.

ERROR(S):

Unable to open a new TCP connection.

POSSIBLE CAUSES:
Maximum number of concurrent TCP sessions configured on this node has been reached. The current setting is: nnn.

SUGGESTIONS:
If this is expected, no action is required. Otherwise, increase the value specified for the following attribute: "TCP Parameters->Active Connection Threshold". This message will not be repeated for this node.

WARNING(S):
TCP sessions opened with this node may experience poor response times.

POSSIBLE CAUSES:
TCP has been configured to create sessions with the following settings:
Receive Buffer: nnn bytes
Receive Buffer Threshold: nnnn
Maximum Segment Size: nnn bytes
Using these settings it is possible to experience a situation when:

1. There is not enough available buffer space to advertise one MSS (note that SWS Avoidance requires this). This causes receiver (this node) to advertise zero available receive buffer size.
2. Sender has more data to send, but cannot send it due to zero remote receiver buffer size.
3. The sender resorts to sending 1 byte at a time as a result of persistent timeout All of this until complete data transfer takes place or when this node advertises a larger buffer will cause data transfer to slow down.

SUGGESTIONS:

1. Set Receive Buffer Threshold to 0.0. This will cause full-window to be advertised for every ACK generated from this node.
2. You can also set the above threshold to abide by the following constraint: threshold > 1.0 - (snd_mss/rcv_buf).

WARNING(S):
TCP session is not being able to transfer all data forwarded by the application. This message will not be repeated.

POSSIBLE CAUSES:
An application sending data over TCP has been configured to send more than nnn bytes over a single TCP connection. However, the current TCP model supports only data transfers of up to mmm bytes per a single TCP connection.

SUGGESTIONS:
Reconfigure applications to send less than nnn bytes over a single TCP connection.

3.15 OSPF-RELATED ENTRIES

BEHAVIOR/RESULT(S):
All OSPF models have been configured to operate in SIMULATION EFFICIENCY mode. In this mode, OSPF on all router nodes will shut down operation after simulation time in seconds. This is the value to which the "OSPF Stop Time" simulation attribute is set. This mode is used to reduce the overall time taken to run the simulation and should be used only when:

1. The state of links and IP router nodes in the network does not change over the course of the simulation.
2. The load on the network as a result of running OSPF is not of interest.

POSSIBLE CAUSE(S):
The "OSPF Sim Efficiency" simulation attribute is set to "Enabled".

SUGGESTIONS:
If either of the two conditions above do not apply to your study, set "OSPF Sim Efficiency" to "Disabled" and rerun your simulation.

SYMPTOM(S):
An inconsistency has been detected when reading the previously exported routing table for the protocol OSPF on this node.

POSSIBLE CAUSE:

1. The export file has been modified after "exporting" it.
2. The scenario for which the routing tables were exported differs from the current scenario.

SUGGESTIONS:

1. Rerun the simulation for this scenario with the simulation attribute "IP Routing Table Export/Import" set to "Export" to recreate the routing tables export file. Then execute the simulation for this scenario with "Export/Import" simulation attribute set to "Import".
2. Run the scenario with the simulation attribute "IP Routing Table Export/Import" set to "Not Used".

SYMPTOM(S):
An OSPF "control" packet was received with invalid header contents:
Neighbor received from: <addr>
Interface received on: <addr>
This packet is being dropped.

POSSIBLE CAUSE:
The router interfaces at the end-points of the link, over which this packet was received, may be configured:

1. in different IP subnets, or
2. in different OSPF areas.

SUGGESTIONS:

For every link on this node, make sure that the interfaces at the end-points of the link are all configured to be in the same IP subnet and the same OSPF area. This message will not repeated for this node for the given interface and remote neighbor.

ERROR:

An inconsistency was detected in the configuration of a virtual link on this node. The error is: <err>. Due to this reason, virtual link nnn will not be set up on this node.

POSSIBLE CAUSE(S):

1. Multiple Areas may not be configured on the router, i.e., the router may not be an Area Border Router (ABR).
2. The router may not have an interface into the configured transit area for the virtual link.
3. The backbone area, Area 0 (0.0.0.0), may have been set as the transit area for the virtual link.

SUGGESTIONS

Make sure that all virtual links are configured correctly:

1. The router must be an ABR, i.e., it must be configured for more than one area.
2. The router must have an interface into the transit area for each virtual link.
3. The transit area for a virtual link cannot be the Backbone area, Area 0 (0.0.0.0).

3.16 IP-RELATED ENTRIES

WARNING(S):
A packet with the wrong format has been\n" received by the IP model.\n"
Rcvd. Packet ID = nnn
Rcvd. Packet Tree ID = nnn
Rcvd. Packet Format = <fmt>
Expected Packet Format is "ip3_dgram"

POSSIBLE CAUSE(S):
The underlying data link protocol(s) have routed the packet incorrectly to IP. Examples of data link protocols are ATM, Frame Relay, and Ethernet.

SUGGESTIONS:
Check the simulation log for messages from data link models that may indicate a possible problem.

ERROR(S):
A local interface could not be found on which the next hop address a.b.c.d. could be reached. The next hop IP address is an interface on the following node: <nodename>

POSSIBLE CAUSE(S):
1. The user-specified value for the "IP Default Route" is incorrect.
2. If this is a routing node and dynamic routing protocols are being used, the next hop address in the dynamic routing table is incorrect.

SUGGESTIONS:
1. Make sure that the value of the "IP Default Route" attr. is part of any one directly connected subnet by correlating it with the subnet masks specified for all the IP interfaces in this node.
2. Debug the configured dynamic routing protocol to find out why (and how) an incorrect next hop address is present in the routing table. Check for messages from these protocols in the simulation log.

ERROR(S):
The user-defined static routing table on this node does not have a route the destination <addr>. The destination IP address mentioned above corresponds to an IP interface in the following node in the network model: <nodename> The IP datagram [ID nnn, Tree ID nnn] is being dropped.

POSSIBLE CAUSE(S):
Same as above

SUGGESTIONS:
Add a route to this destination in the "Internal Routing Table" sub object of the "IP Routing Information" attribute on this node.

WARNING(S):
The default route <route> specified for this node is invalid.

REASON(S):
The specified default route is not directly connected to this node. Default routes specified should be within one ip hop from the node. i.e., it should be in the same IP subnet as one of the interfaces of this node.

SUGGESTION(S):
Specify a valid default route.

RESULT(S):
The default route specification will be ignored. NOTE: This message will not be repeated.

ERROR(S):
Unable to route packet to the destination a.b.c.d.

REASON(S):
While doing a recursive lookup to find the next hop to reach this destination, a loop was encountered.

POSSIBLE CAUSE(S):

1. Routing Protocol misconfiguration
2. Incorrect static routes

SUGGESTION(S):
Export the routing table of this node and try to find the erroneous route(s).

WARNING(S):
The following routing table entry inserted into the common route table by the <protocol> routing protocol did not specify the interface through which the next hop can be reached.

Destination: a.b.c.d.
Subnet Mask: a.b.c.d.
Next Hop: a.b.c.d.
The next hop is in the same subnet as interface# nnn in the ip interface table.

RESULTS(S):
The interface to reach the next hop will be set as nnn.

SUGGESTIONS(S)

1. If this is a standard routing protocol, please contact OPNET Technical support.
2. If this is a custom routing protocol, make sure that the port_info element of the route is specified correctly while inserting the route into the common routing table.

WARNING(S):
A route computed by OSPF to a destination will not be redistributed to other routing protocols that may have been configured on this router node. The destination IP address is a.b.c.d.
The destination IP address above corresponds to an interface on the following node: <nodename>].

POSSIBLE CAUSE(S):
This is a destination connected to this router via point-to-point link technology. OSPF associates a mask of 255.255.255.255 with such destinations, and uses this mask value when finding the best match route to a destination during route table lookup.

SUGGESTIONS:

1. If there are no other routing protocols configured on this router apart from OSPF, ignore this message.
2. If this destination corresponds to a router interface, ignore this message.
3. If this destination corresponds to a host interface, i.e., this is a host route, then:
    a. Connect the host to this router with non point-to-point link technology.
    b. Do not use OSPF as the routing protocol on this router's interface that connects to the host. NOTE: This message will not be repeated.

WARNING(S):
The IP packet (ID nnn, Tree nnn) is being dropped because its TTL field decrements to zero.
Interface Received: a.b.c.d
Packet Destination: a.b.c.d

POSSIBLE CAUSE(S):

1. The network model has IP routing tables with loops.
2. The optimal routed path taken by this packet does indeed have more than 32 hops.
3. This might be a directed broadcast packet whose TTL was set to 1 by the broadcasting router to prevent the packet from being routed.

SUGGESTIONS:

1. If hand configured, make sure that the "Internal Routing Table" does not have any route loops.
2. If this is a directed broadcast packet, find out why it was delivered to this node. The most probable cause for this is that interfaces belonging to the same lower layer network do not have the same IP Network Address.

WARNING:
The IP Routing Table has lost its route to IP network (a.b.c.d) (e.f.g.h). using <protocol>. It <does/does not> have a backup using another protocol.

POSSIBLE CAUSES:
If a node or link has been set to fail this is expected behavior. If this isn't the case, check for log messages within the specific routing protocol.

WARNING:
Simulation encountered some IP network addresses that are not registered in the global IP table. Information about these networks has been logged in the following file: <proj>-<scen>-ip_addr_err_log.gdf. The global IP table is maintained for a faster lookup, and is mainly used by RIP and IGRP. This table contains destination addresses based on their classful boundaries.

REASON(S):

1. None of the IP interfaces in the current network model belong to the unregistered networks. This is possible with router configuration import where only part of the network is being imported. The IP networks that trigger this warning belong to the part of the network that has not been imported. But static routes to these networks are configured on the imported routers and are redistributed to other routing protocols.
2. The networks that trigger these error messages are summary addresses that are being redistributed into RIP, IGRP, or EIGRP.
3. The networks are subnetted with VLSM and are being redistributed into classful protocols like RIP, IGRP, and EIGRP.

RESULT(S):
In all the above cases the routing tables that are built by the routing protocols and the forwarding tables may not contain these addresses. This will be specifically true on routers that use RIPv1 or IGRP to build the forwarding tables. Note this will not have any significant impact on the routing studies that you may perform.

ERROR:
An ARP module in the network encountered an IP packet whose next hop address could not be mapped to a MAC layer address. Information about all such addresses has been logged in the following file: <proj>-<scen>-ip_addr_err_log.gdf

POSSIBLE CAUSE(S):

1. No interface with these IP addresses exists in this network.
2. The interfaces with these addresses do not run a compatible MAC protocol.

SUGGESTION(S):

1. Make sure that these IP addresses exist.
2. Make sure they belong to the correct MAC protocol.

RESULT(S):

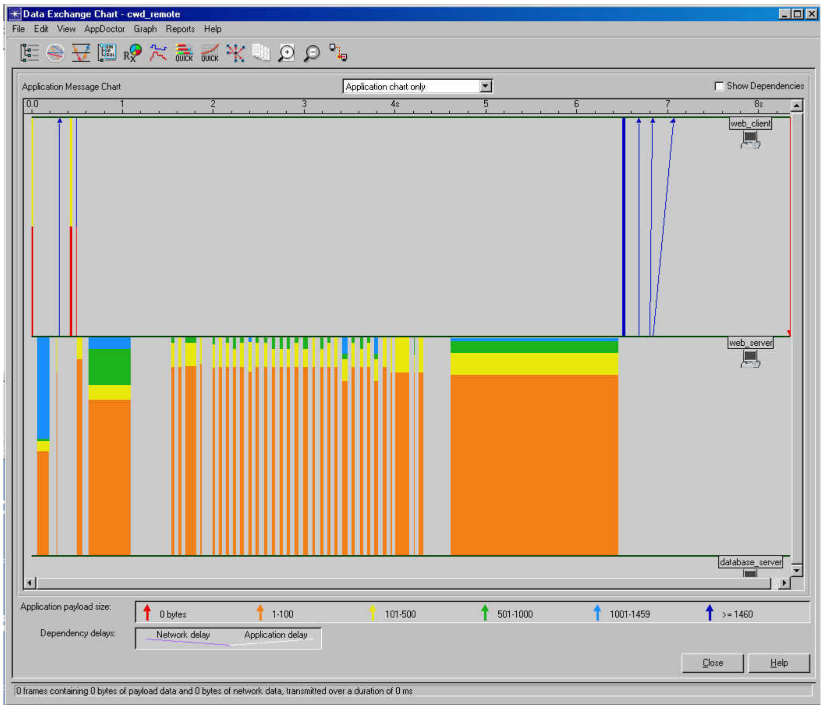
1. These packets will be dropped.

APPENDIX C – Example Run Matrix

Excursion/Scenario	Purpose	Project	Scenario	Wkstn	Date	BKGD Traffic %	Duration (sec)	Real Time (sec)	Seed	# Sim Log Events	# Sim Events	Archive	Comments
BASELINE	Verify the "clean" simulation. Fix simulation log entries and verify statistics collection.	CE_STANDALONE	CWAN	Young	12/1/03	10	3600	1260	128	17	2	C:\Studies\CE04	
VOICE	Verify simple voice profiles (5 minute conversations between NETOP and Italy LAN)	CE_STANDALONE	VOICE	Smith	12/1/03	10	3600	1260	128	17	2	C:\Studies\CE04	
VIDEO	Verify simple voice profiles (5 minute conversations between NETOP and Italy LAN)	CE_STANDALONE	VIDEO	Turner	12/1/03	10	3600	1260	128	17	2	C:\Studies\CE04	
DCTS	Verify simple DCTS profile between NETOPs and Italy LAN	CE_STANDALONE	DCTS	Brown	12/1/03	10	3600	1260	128	17	2	C:\Studies\CE04	

APPENDIX D– Example Imported Application Profile

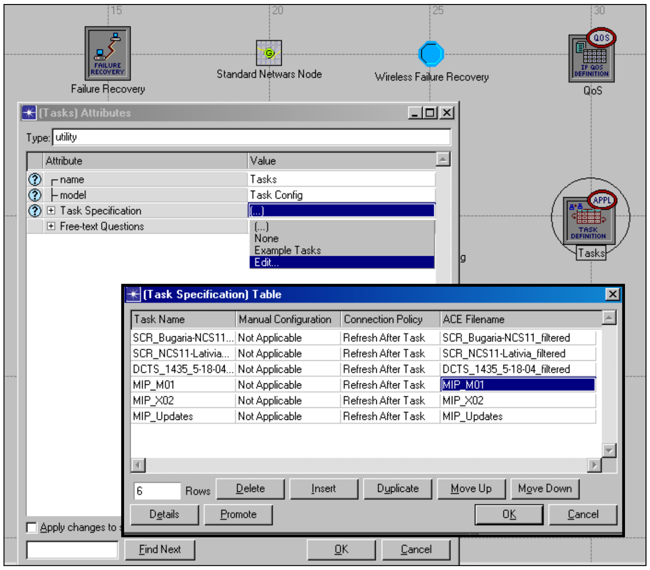
The data used in this example was captured during an exercise using and was filtered and examined in OPNET's Application Characterization Environment (ACE) tool. The primary role of ACE for this effort was to develop application profiles using the captured data. These profiles contain all of the application transfers between logical "tiers" associated with an application. Once the applications were filtered in ACE, they were imported directly into NETWARS. Figure D-1 shows an example of an ACE trace associated with a simple web based application that pulls data from a common database.



0 frames containing 0 bytes of payload data and 0 bytes of network data, transmitted over a duration of 0 ms

Figure D-1
Example ACE Trace – Transactional View

When an application is imported into NETWARS, the software automatically creates an associated task in the "Tasks" utility (see Figure D-2). The task specification is how NETWARS tracks imported applications.



(Tasks) Attributes

Type: utility

Attribute	Value
name	Tasks
model	Task Config
Task Specification	(...)
Free-text Questions	

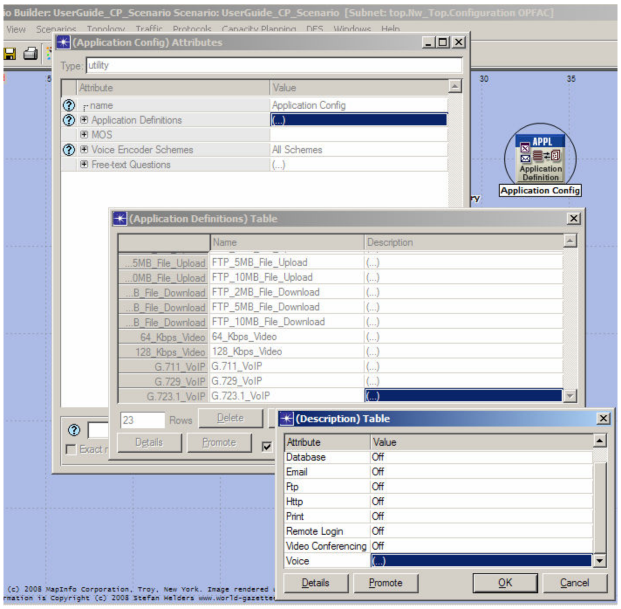
(Task Specification) Table

Task Name	Manual Configuration	Connection Policy	ACE Filename
SCR_Bugaria-NCS11...	Not Applicable	Refresh After Task	SCR_Bugaria-NCS11_filtered
SCR_NCS11-Lativia...	Not Applicable	Refresh After Task	SCR_NCS11-Lativia_filtered
DCTS_1435_5-18-04...	Not Applicable	Refresh After Task	DCTS_1435_5-18-04_filtered
MIP_M01	Not Applicable	Refresh After Task	MIP_M01
MIP_X02	Not Applicable	Refresh After Task	MIP_X02
MIP_Updates	Not Applicable	Refresh After Task	MIP_Updates

6 Rows

Figure D-2
NETWARS Task Utility

Imported tasks are automatically assigned as an application inside the "Application Configuration" utility (see Figure D-3) and their default attributes are usually sufficient. However, it is here users must first turn on NETWARS application models and configure the attributes to represent the desired traffic (see the next section, Voice over Internet Protocol for an example).



o Builder: UserGuide_CP_Scenario Scenario: UserGuide_CP_Scenario [Subnet: top.Nw_Top.Configuration OPFAC]

(Application Config) Attributes

Type: utility

Attribute	Value
name	Application Config
Application Definitions	(...)
MOS	
Voice Encoder Schemes	All Schemes
Free-text Questions	(...)

(Application Definitions) Table

	Name	Description
...5MB_File_Upload	FTP_5MB_File_Upload	(...)
...0MB_File_Upload	FTP_10MB_File_Upload	(...)
...B_File_Download	FTP_2MB_File_Download	(...)
...B_File_Download	FTP_5MB_File_Download	(...)
...B_File_Download	FTP_10MB_File_Download	(...)
64_Kbps_Video	64_Kbps_Video	(...)
128_Kbps_Video	128_Kbps_Video	(...)
G.711_VoIP	G.711_VoIP	(...)
G.729_VoIP	G.729_VoIP	(...)
G.723.1_VoIP	G.723.1_VoIP	(...)

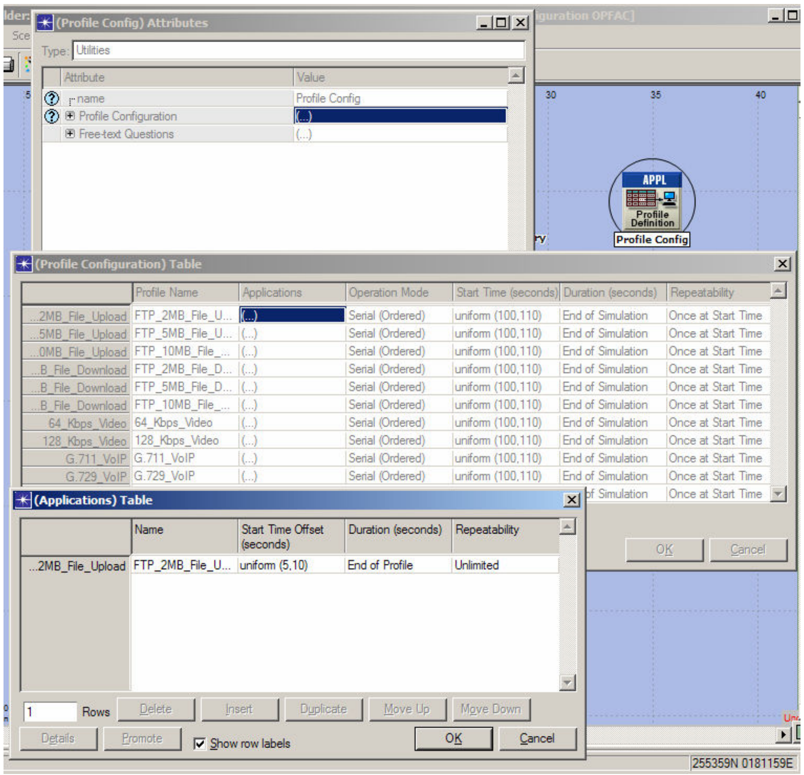
23 Rows

(Description) Table

Attribute	Value
Database	Off
Email	Off
Ftp	Off
Http	Off
Print	Off
Remote Login	Off
Video Conferencing	Off
Voice	(...)

Figure D-3
Application Configuration Utility

Lastly, applications must be assigned to profiles to describe how they will behave during the simulations (see Figure D-4). Each profile must have an associated application(s), which are specified by the “applications” attribute. Multiple applications can be assigned to a profile. However, users must consider the order in which they are entered into the utility. For example, if the “Operation Mode” is set to “serial (ordered)” the applications must be listed in the correct order.



Attribute	Value
name	Profile Config
⊞ Profile Configuration	(...)
⊞ Free-text Questions	(...)

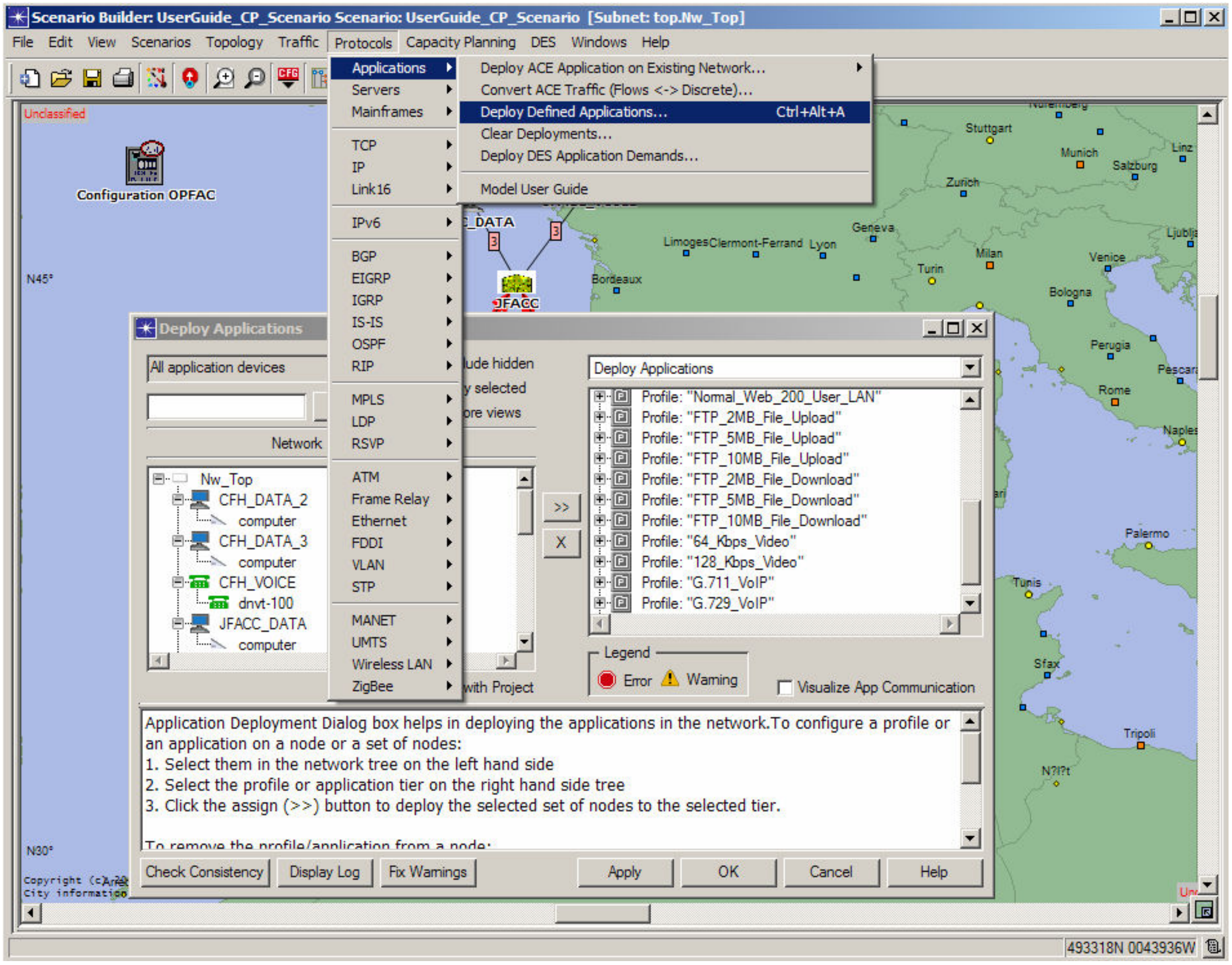
	Profile Name	Applications	Operation Mode	Start Time (seconds)	Duration (seconds)	Repeatability
...2MB_File_Upload	FTP_2MB_File_U...	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
...5MB_File_Upload	FTP_5MB_File_U...	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
...0MB_File_Upload	FTP_10MB_File_...	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
...B_File_Download	FTP_2MB_File_D...	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
...B_File_Download	FTP_5MB_File_D...	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
...B_File_Download	FTP_10MB_File_...	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
64_Kbps_Video	64_Kbps_Video	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
128_Kbps_Video	128_Kbps_Video	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
G.711_VoIP	G.711_VoIP	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
G.729_VoIP	G.729_VoIP	(...)	Serial (Ordered)	uniform (100,110)	End of Simulation	Once at Start Time
					f Simulation	Once at Start Time

	Name	Start Time Offset (seconds)	Duration (seconds)	Repeatability
...2MB_File_Upload	FTP_2MB_File_U...	uniform (5,10)	End of Profile	Unlimited

Rows: 1

Figure D-4
Profile Configuration Utility

Lastly, the profiles are assigned to locations within the architecture using the NETWARS application deployment editor (see Figure D-5). End user devices (on the left) are assigned to the appropriate application tier (on the right) using the “>>” button. They can also be deleted by using the “X” button.



Application Deployment Dialog box helps in deploying the applications in the network.To configure a profile or an application on a node or a set of nodes:
1. Select them in the network tree on the left hand side
2. Select the profile or application tier on the right hand side tree
3. Click the assign (>>) button to deploy the selected set of nodes to the selected tier.

To remove the profile/application from a node:

Figure D-5
NETWARS Application Deployment Editor

NOTE: ACE does not correctly characterize User Datagram Protocol (UDP) traffic. ACE handles UDP traffic in the same manner as Transport Control Protocol (TCP) traffic, which introduces a considerable amount of additional over-head. This issue requires UPD applications to be duplicated in NETWARS instead of importing captured data files. This can accomplished using the NETWARS application models to represent UDP applications (i.e., VTC and VoIP).

APPENDIX E – Abstract Modeling

The analysis of large, complex DoD communication networks such as the GiG is an increasing concern, but such networks are difficult to analyze because they are topologically complex, highly non-linear in their responses, and inherently unbounded (i.e., no node in the network has global knowledge of the entire network). As a result of the sheer number of components involved in the large-scale systems, it has always been a sad irony that modelers and management have been relegated to the use of limited tools to represent the behavior of these entities. The irony stems from the fact that, despite the existence of accurate models of most of the entities, they are traditionally not used for large-scale simulations; primarily because of the increased runtime that they bring to the system. Consequently, it is impossible for anyone, whether they are users, planners or analysts, to focus on an entire network at once. Nor is it possible to present a graphical model of a large, complex system on a single sheet of paper. So our modeling techniques must allow us to build the abstract of an architecture that portray individual parts of a communications network. Analysts and Planners must remember the models are built for the following reasons:

1. To focus on important system features while downplaying less important features.
2. To discuss changes and corrections to the user's requirements at a low cost and with little risk.
3. Lastly, to verify that we understand the user's environment and have documented it in such a way that system analysts and planners can build the systems.

Therefore, to support analysis of these networks this appendix will describe how to develop statistically valid abstract networks for analysis and, as an example of their use, applying them to the simulation of network performance. This section will illustrate the construction of network topologies using NETWARS. The goal of communication modeling is to ultimately prove that a message has been sent and received. This is extremely important in C4I networks where commands and status must be issued and a response is required.

In order to easily and efficiently understand the characteristics of large networks, focus must first be placed on the construction of abstract models that exhibit statistical properties comparable to the actual network. This approach is done in phases to characterize the behavior of the network, which are outlined in the subsequent sections below.

3.17 INFRASTRUCTURE DEVELOPMENT

Critical links and devices are identified that support the goals and objectives of the study. When conducting any type of study or performing an analysis it is very important to identify the goals and objectives of the study. This process will help the analyst identify critical points of the architecture whether they are links and/or devices. Once these devices have been identified they must be connected. Build the model incrementally by adding nodes one at a time remembering that the finished product must support the goals and objectives of the study.

3.18 TRAFFIC DEVELOPMENT

Develop necessary traffic to traverse the abstract architecture. For example, if your goal is to analyze a segment of a network you only need to develop traffic for that segment not the entire network.

3.19 ACCURACY, COMPLEXITY AND LEVEL OF DETAIL

Scenario accuracy can be thought of as addressing the issue of how a well a scenario represents the accurateness of the network. It is tied to model validation activities, and as such, it is a quantifiable concept.

The level of detail relates more to how the scenario is represented in NETWARS. Specifically, to what level are communication device models of the network modeled in NETWARS. Finally, complexity is related to the computational aspects of executing the code that represents a scenario's behavior within the NETWARS environment. These three concepts are closely related, but their appropriate use has been confusing. Below are some assertions regarding how these concepts are different, and how they relate to each other:

Assertion 1: Increased level of detail does not imply increased accuracy.
Assertion 2: Increased complexity does not imply increased accuracy.
Assertion 3: Increased level of detail usually does imply increased complexity.
Assertion 4: Complexity is directly related to computer runtime.

The assertions show that accuracy does not appear to be inexorably linked to complexity or to level of detail. Therefore on the surface, it would seem reasonable to assume that, if a scenario's complexity can be reduced (which implies, but does not demand, that we reduce the scenario's level of detail), the scenario will produce valid results. The assertions also show that reducing a model's complexity reduces its associated runtime. Consequently the goal of scenario abstraction is to sufficiently reduce the complexity of a scenario, without experiencing too great a loss in accuracy.

Individual Scenario Abstraction techniques vary tremendously. It is not an exact science however, a current categorization of scenario abstraction techniques have grouped these techniques into three broad approaches; scenario boundary modification, model behavior modification, and scenario form modification. A description of these approaches to scenario abstraction is listed below:

The simulation of a battalion's ability to communicate from Theater back to Headquarters in CONUS has the potential to offer a tremendously computationally complex scenario. Consider the need to simulate on-the-fly: Network congestion due to background from the entire network, computational limitations due of the computing environment etc. Especially pertinent is the potential impact of terrain on wireless and/or satellite traffic. ·To illustrate abstraction opportunities, techniques from each of the three model abstraction categories can be employed to reduce the complexity associated with simulating the reach-back communication from a battalion in theater back to CONUS. The first abstraction is a scenario boundary identification, which involves changing variables associated with the scenario. Simulation execution time can be significantly reduced by decreasing communication device models associated with the scenario. For example instead of looking at the network objects behind each satellite terminal an analyst can roll-up the architecture and associated traffic to the source and destination satellite terminals to examine performance.

A second abstraction could take the form of a model behavioral modification. These techniques involve the changes to the internal elements of communication device models. The behavior of a model can be greatly improved by manipulating attributes associated with that model. Manipulating the impact of model attributes can generate abstractions of a detailed communication device model. Although the resultant performance may not be as accurate to the n^{th} decimal as the one produced from the baseline model, it does however provide traceability of traffic performance. The absence of this traceability is one the primary factors that adversely impacts the credibility given to the results of large-scale scenarios.

Lastly, there exists an opportunity to modify the model form (i.e., replace the model with a surrogate model). An easy and often used technique is to utilize surrogate models to replace devices that are not within the NETWARS model library. The use of these simple rules results in abstract networks that reflect the statistical properties of the actual networks.

APPENDIX F – Acronyms

ABR	Area Border Router
ACK	Acknowledgements
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
C4	Command, Control, Communications, Computer
C4I	Command, Control, Communications, Computer, and Intelligence
CADM	Core Architecture Data Model
CDM	Communication Device Model
COTS	Commercial Off The Shelf
DISA	Defense Information Systems Agency
DoD	Department of Defense
EIGRP	Extended Interior Gateway Routing Protocol
EPLRS	Enhanced Position Location Reporting System
FR	Frame Relay
GB	Giga-Byte
GHz	Giga-Hertz
GOE	Generic Organization Editor
GUI	Graphical User Interface
HDD	Hard Disk Drive
HL	High Level Architecture
HTML	Hyper-Text Markup Language
IER	Information Exchange Requirement
IGRP	Interior Gateway Routing Protocol
IP	Internet Protocol
IPR	In-Progress Report
JCSS	**Joint Communications Simulations System**
JTF	Joint Task Force
JS	Joint Staff
LAN	Local Area Network
MAC	Medium Access Control
MOE	Measure of Effectiveness
MOP	Measure of Performance

MSE	Mobile Subscriber Equipment
M&S	Modeling & Simulation
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NETWARS	NETwork WARfare Simulation
OPFAC	OPerational FACility
OPLAN	Operation Plans
OPNET	Optimum Network
OSPF	Open Shortest Path First
OV	Output Vector
PNNI	Private Network-to-Network Interface
RAM	Random Access Memory
RIP	Routing Information Protocol
SAR	Satellite Access Request
SATCOM	Satellite Command
SCM	Scenario Conversion Module
SDF	Simulation Description File
SE	System Element
SINCGARS	Single Channel Ground & Airborne Radio
SIPRNET	Secure Internet Protocol Routing Network
STEP	Standardized Tactical Entry Point
TCP	Transmission Control Protocol
TIREM	Terrain Integrated Rough Earth Model
UDP	User Datagram Protocol
URC	Unit Relationship Code
VTC	Video Teleconferencing
V&V	Verification & Validation
XML	Extensible Markup Language
WAN	Wide Area Network