REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
30–04–2005	Final	20 April 2004 — 30 April 2005

4. TITLE AND SUBTITLE

0003AC - FINAL TASK ORDER TECHNICAL REPORT

Investigations Into The Application Of NMS Tools To Modeling The Global Information Grid (GIG)

5a. CONTRACT NUMBER
MDA972–01–D–0005

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Doshi, Bharat, and Cole, Robert G.

5d. PROJECT NUMBER
N/A

5e. TASK NUMBER
0037

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Johns Hopkins University Applied Physics Laboratory
Power Projection Systems Department
11100 Johns Hopkins Road
Laurel, MD 20723-6099

8. PERFORMING ORGANIZATION REPORT NUMBER
N/a

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA/ITO
3701 N. Fairfax Dr., Arlington, VA 22203–1714

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
N/A

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A (Approved for Public Release, Distribution Unlimited) - #6106

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the authors and should not be construed as an official U.S. Government position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

This report documents our work in support of transition of the capabilities and tools developed within the Defense Advanced Research Project Agency (DARPA) Information Program Technology Office (IPTO) Network Modeling and Simulation (NMS) Program to the greater Department of Defense (DoD) community. We chose to use the efforts to design and architect the Global Information Grid (GIG) as a test case. We then identified a set of performance studies of interest to GIG engineers and architects, compared the set of tools and capabilities derived within the DARPA NMS program against the needs of these performance studies, and identified a software architecture (based upon these and related tools) for a reusable GIG simulation platform. In the process we identified a set of gaps in tool sets that require further work, i.e., capabilities software development and integration.

15. SUBJECT TERMS

Technology Transfer, GIG Simulation Model, Tools Assessment, BGP Performance.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Bryce (Andy) Thompson
U	U	U	SAR	37	19b. TELEPHONE NUMBER *(include area code)* (240) 228–0708

0003AC - FINAL TASK ORDER TECHNICAL REPORT *

INVESTIGATION INTO THE APPLICATION OF NMS TOOLS TO MODELING THE GLOBAL INFORMATION GRID (GIG)

---

30 April 2005

Submitted to:	DARPA Information Technology Office 3701 N. Fairfax Drive Arlington, VA 22203-1714
Submitted by:	Johns Hopkins University Applied Physics Laboratory 11100 Johns Hopkins Road Laurel, MD 20723-6099
Principal Investigator:	Dr. Bharat Doshi, 240-228-0625
Project Manager:	Dr. Robert G. Cole, 240-228-6951
Program Manager:	Andy Thompson, 240-228-0708

Prepared under Contract:	MDA 972-01-D-0005
Task Order:	0037
Awarded:	20 April 2004

Contents

1 EXECUTIVE SUMMARY

This report documents our work in support of transition of the capabilities and tools developed within the Defense Advanced Research Project Agency (DARPA) Information Program Technology Office (IPTO) Network Modeling and Simulation (NMS) Program to the greater Department of Defense (DoD) community. The method we followed to accomplish this task was to choose a test project related to modeling and simulation analysis of interest to the DoD in order to demonstrate the utility of the NMS tools and capabilities. We chose to use the efforts to design and architect the Global Information Grid (GIG) as a test case. We then identified a set of performance studies of interest to GIG engineers and architects, selected a set of tools and capabilities derived within the DARPA NMS program based upon the needs of these performance studies, and identified a software architecture (based upon these and related tools) for a reusable GIG simulation platform. We ran through an initial test case to validate our process and enhance our understanding of the future work required to better integrate NMS developed tools into an integrated GIG simulation tool. In the process we identified a set of gaps in the selected tool sets that require further work, i.e., capabilities software development and integration (as identified in Section 10). We socialized our work and the capabilities of the NMS program to various organizations within the DoD (as identified in Section 9). We concluded our efforts with a set of recommended work items to further encourage the transition of NMS tools and capabilities into the greater DoD community.

2 OVERVIEW

On 5 March 2004, Johns Hopkins University Applied Physics Laboratory (JHU/APL) submitted a proposal, i.e., "Network Modeling and Simulation (NMS) Program", to support the Defense Advanced Research Project Agency (DARPA) Information Program Technology Office (IPTO) NMS Transition initiative. Task Order 0037 was awarded 20 April 2004 with a period of performance through 19 February 2005. Amendments 1 and 2 were no cost extensions to 31 March 2005 and 30 April 2005 respectively. This document is the Final Report associated with this project. The goal of this final report is to summarize our efforts in supporting the transition of the DARPA IPTO NMS Program to the greater DoD community.

Our approach to support this transition effort was to take the ongoing work in the DoD to develop the GIG network as a test case and to identify a set of performance modeling studies of relevance to these efforts. Then, we investigated the application of NMS tools and techniques toward the resolution of these performance studies. At the time of the award of this contract, JHU/APL had an ongoing Internal Research and Development (IR&D) project, which had been developing preliminary models of the GIG and some potential areas of study to aid GIG designers. Further, JHU/APL was actively participating in several of the GIG Working Groups involved with GIG Routing, End-to-End Architecture, Quality of Service (QoS), and Network Management. On 2 August 2004, a "Jump Start" effort was initiated to pull together work and ideas of the various JHU/APL personnel participating in these GIG performance, design, and architecture working groups and develop a plan for this project.

The methodology developed for this transition project follows: First, a desirable set of performance studies for the GIG designers was identified. This set of studies was used to define a set of requirements for a reusable network simulation model of the GIG, and these drove a selection of the NMS tools forming the basis of the reusable GIG network simulation tool. The selection of the NMS tool set a) identified tools useful for the various performance studies related to the GIG, and b) identified gaps in the selected NMS tool set. Next, a topology model of the GIG was developed upon which to base these performance studies. Two views of the GIG topology were developed. The base topology would potentially assist in developing a re-usable GIG simulation platform and allow a consistent set of topologies to compare results across the different performance studies. Then a test case was run through based upon one of the proposed performance studies, i.e., BGP performance within the GIG infrastructure. Running through this test case resulted in a more meaningful investigation of the NMS tools set and improved our recommendations for follow on work activities to transition NMS capabilities to the DoD. In the process of executing the test case, additional capabilities were developed in order to automate a) the generation of the GIG topology in terms of the simulation tool's configuration language, b) the analysis of the simulation logs and trace files, and c) the generation of performance metric reports. We socialized our work results and recommendations with various DoD organizations in order to stimulate interest in the transition of the NMS tools into the DoD Modeling and Simulation (M&S) community.

We participated in the final two Principal Investigators (PIs) meetings. These meetings were used as a vehicle for us to develop a broad appreciation of the breadth and depth of the activities performed under the NMS program. These meetings served to socialize our approach and methodology to help promote the transition of these capabilities into the DoD M&S community. Specifically, the final PI meeting was used to gain concurrence for our methodology from the program manager and PIs associated with the NMS program. We participated in the Navy's

M&S Technical Interchange Meetings (TIM), set up by the DARPA NMS Transition Team Lead, from the Navy's SPAWAR organization. Other meetings were set up in order to socialize our findings and work on this project as discussed in Section 9 of this report.

In the remainder of this report, we go into the details of our approach, our test case analysis and findings, and our efforts to socialize our work and findings. Section 3 defines the task objectives. Section 4 identifies some of the technical difficulties in the execution of this project. Section 5 details the methodology taken in the execution of this project. Section 6 presents the technical results of our work and discusses a "Test Case" where we worked through the necessary development steps to better flesh out missing details in our technical approach. Section 7 lists our important findings and conclusions. Section 8 describes the software development associated with our work. Section 9 documents the technical exchange meetings, contacts, and discussions we have had to aid in the socialization of our work in order to encourage transition into the DoD M&S community. Section 10 discusses follow on work related to continuing the transition effort.

3 TASK OBJECTIVES

The project objectives were to:

- investigate the application of the tools and techniques developed within the DARPA IPTO's NMS program to networking applications within the DoD and
- aid in the transition of these tools into the DoD community.

The scope of our project did not allow for an investigation of all the different tools, methods and techniques developed within the NMS program. So instead a more focused investigation was performed. In order to accomplish these objectives, we investigated the NMS developed tools in the context of developing a reusable simulation platform for GIG performance studies. We demonstrated the utility of the reusable GIG simulation model by executing an initial test case and socializing our work with various DoD organizations. In the process, we developed associated configuration tools and identified gaps in the NMS tool set. This led to recommendations for future development activities to aid in better integration of these NMS capabilities.

4 TECHNICAL PROBLEMS

The contract awarded JHU/APL was to further the transition activities of the DARPA IPTO NMS project. The JHU/APL award came at the tail end of the NMS project, which was begun in 2000. Most of the work within the NMS project had been completed by the time of this award. The approach to achieving the objectives of this project was to demonstrate the value of NMS tools by applying a set of the tools against the resolution of performance studies of value to GIG designers. In the process, a set of recommendations for future integration and development work related to NMS support tools and capabilities resulted. Some of these recommendations for future work may have been included in the NMS program if this analysis had been performed in an earlier phase of the program.

5 GENERAL METHODOLOGY

In this section, we detail the methodology followed during the execution of the task related to this project. The approach to completion of the project was to demonstrate the value of NMS tools and capabilities by a) assembling a subset of these tools into a reusable GIG simulation model at JHU/APL, b) executing a performance study on the GIG simulation model showing the utility of the NMS tools and capabilities, and c) socializing our efforts with organizations within the DoD. We followed a classical, systems engineering, top-down approach resulting in the following phases:

- *Performance Study Definition* - a set of performance studies were defined, that add value to the activities of the GIG Working Groups and the DoD organizations and services in designing the GIG and it's respective component networks.
- *GIG Characterization* - a consistent and realistic model of the GIG was developed which would lend itself to large scale simulation efforts. This model is consistent with GIG plans across the various branches of the DoD.
- *NMS Tools Assessment* - with a set of performance studies to perform, the tools and techniques from the NMS program were assessed with respect to their utility in supporting the desired performance studies.
- *Test Case* - the transitioning of NMS capabilities is necessarily a longer activity extending beyond the scope of the current DARPA NMS project. However, it was useful to work through a "Test Case" in order to validate the approach chosen in this project and to demonstrate the utility of NMS capabilities in addressing DoD related network M&S studies. This test case also began to address the issue of simulation tools Verification and Validation (V&V) and refined our understanding of future work recommendations.
- *Tools Enhancements* - several enhancements of the existing tools were be identified in order to improve the applicability of the NMS tools to DoD programs.
- *Socialize Efforts with DoD* - in order to help in marketing the NMS developed capabilities, we set about socializing our work with interested parties within the DoD, including Office of the Secretary of the Defense - Network Information Infrastructure (OSD-NII), U.S. Navy, U.S. Army, the DoD's High Performance Computing Modernization Program (HPCMP) and the U.S. Air Force.

6 TECHNICAL RESULTS

In this section, we present the technical results of our effort. Following through with our methodology described above, we first developed a set of performance studies for the GIG designers. We then developed various models of the GIG for simulation modeling. The GIG models were developed in such a way that we could parameterize their descriptions and build software to automatically generate simulation configuration as input into the simulation tools. We then performed a selection of the available NMS tool sets, based upon their applications to the objective studies. Finally, we ran through a study test case in order to better flesh out our methodology and demonstrate the value of NMS capabilities to the DoD community. We describe this work below.

6.1 Performance Study Definition

Here we discuss the performance studies which were identified as a useful set of Modeling and Simulation (M&S) studies for the GIG designers and architects. The intent of defining these studies was to demonstrate the NMS capabilities against a useful set of M&S activities for the DoD. In this way, requirements for the tools selection would be driven by the end-users' needs, i.e., the DoD. The following set of performance studies were proposed for the assessment of the NMS capabilities:

BGP Stability and Scale The GIG is relying upon the Border Gateway Protocol version 4 (BGP) [2] to meet various routing demands in a multi-domain and large scale network. This set of simulation studies focuses on the scalability of the alternative inter-domain routing architectures under consideration by the GIG designers. This work also includes convergence time studies to investigate BGP routing behavior under various failure scenarios and under various network mobility assumptions. Based upon the NMS tools and capabilities (see the discussion below in Section 6.3 below), we proposed using the NS2 [16], PDNS [17], and the BGP++ [4] tools for this study. We discovered that some integration work is required to support these studies.

QOS Enhancements The GIG will comprise of various and diverse technologies supporting a broad range of applications with diverse network performance requirements. In addition, the GIG may pioneer the implementation of Multi-Level Precedence and Preemption (MLPP) capabilities. In order to accomplish this, protocol modifications and enhancements of networking protocols are required, including QoS enhancements to routing protocols, resource reservations through signaling, source routing, tunnel establishments, etc. New protocols may need development. These will require supporting M&S efforts. A specific example of studies of this type is to investigate modifications to BGP to carry QoS related metrics. An M&S study on BGP QoS enhancements would require fairly extensive code integration the above identified tool set. Further, additional NMS capabilities would be useful. Specifically, this study would require better integration of BGP++, NS2 and the IFFM models [14] from the DARPA NMS project. This would also require code modification of the BGP++ tool.

End-to-End Application Modeling This study focuses on the performance of various applications carried over the GIG. Due to the rather broad range of network technologies

supporting the overall connectivity within the GIG, this study focuses on the range of end-user perception of performance expected at various application attachment points to the GIG. Potential applications include Voice over IP (VoIP), Situational Awareness (SA), Mobility Alert (MA), Sensor-Fusion-Shooter, and Target Assessment (TA) applications. NMS tools related to application and network traffic modeling would be useful for this type of studies.

Information Assurance The platform and tool sets from the NMS program could be applied to Information Assurance (IA) investigations of the GIG. Overlay security systems and applications could be run over a GIG simulation in order to better understand the performance and effectiveness of various security protocols and systems prior to deployment in the GIG. These studies need to be fleshed out in terms of the nature of the attacks and mitigation technologies under study for potential GIG deployments. Example studies could include the performance of High Assurance IP Encryptor (HAIPE) discovery and routing protocols in GIG deployments, worm infection spreading and mitigation technology effectiveness and prevention, and attacks against the routing infrastructure.

It was not the intent of this project to execute these studies. Instead we used these studies as a guide in our selection of a subset of NMS developed tools. We then used these tools to demonstrate utility of NMS capabilities in the execution of a test case.

6.2 GIG Characterization

In this section, we discuss the work to characterize the topology and scale of the GIG. The purpose of this work was to characterize the GIG as an end-to-end system for M&S purposes. Hence, a goal of this characterization was to parameterize the topology so that automated generation of simulation configurations were possible. This placed some restrictions on the level of detail placed in the models.

The U.S. Federal Government's GIG network is currently under design and development. Hence, the topology decisions made within our simulation studies include assumptions on a future network topology still in flux. It also includes assumptions to simplify the initial simulation configurations and allows for a simple parameterization of the topologies. However, the intent of the topology described herein was to provide a reasonable representation of the future GIG with enough complexity to capture the performance details under investigation. We were *not* trying to create a detailed model of constituent programs. Instead, we looked to abstract the capabilities of constituent programs to develop an appropriate model to analyze end-to-end application and inter-domain routing performance; treating the GIG as an overall system. The topology we chose followed closely the work in the GIG Routing Working Group (GRWG).

Two views of the GIG architecture were addressed: one corresponding to near-term evolution (prior to 2010) and another long-term (after 2012 and the deployment of the Transformational Satellite (TSAT)). Both views are idealized and abstracted from the views of the constituent programs. The deployment of the TSAT system represents a dramatic change in the routing architecture within the overall GIG network. Hence we chose the deployment of the TSAT system as the demarcation between our near-term and long-term architectures.

A key architectural construct for the GIG is the *Black Core.* In the black core, all traffic has been encrypted and can be treated as unclassified. In particular, classified IP addresses are not visible in the black core. Red networks - networks that use classified IP addresses - connect

to the black core through HAIPE devices that carry the classified traffic in Encrypted Secure Protocol (ESP) tunnels. From a routing perspective, we assumed that classified networks are stub networks and that all routing information is aggregated into route-able prefixes at the point of attachment to the GIG.

Figure 1 shows an example high level, near-term topology chosen as our baseline for the set of simulation studies previously discussed. The GIG is composed of various networking domains managed by various organizations within the DoD. These include the GIG Bandwidth Expansion (GIG-BE) and GIG End User (GIG-U) networks, Teleport, the Transformational Satellite (TSAT) network, the Automated Digital Network System (ADNS), ADNS Users (ADNS-U), and the Army's War-fighter Information Network - Tactical (WIN-T). The solid lines in the figure are meant to indicate wired-lines; while the dashed lines indicate wireless connectivity, e.g., satellite links. The core networks, i.e., GIG-BE, ADNS, CITS, and Teleport, are comprised of very high speed communications facilities. Additionally, GIG-BE plans on supporting IP and Multi-Protocol Label Swapping (MPLS) based services for transport. Other GIG networks may provide MPLS transport as well, in addition to IP transport. Our current GIG models capture the dynamic nature of the satellite connectivity down to the tactical environments, e.g., ADNS-U (battleship groups) and WIN-T MANETS (Army Unit of Action deployments). Our development of this GIG topology model for M&S purposes is more fully described in [8].

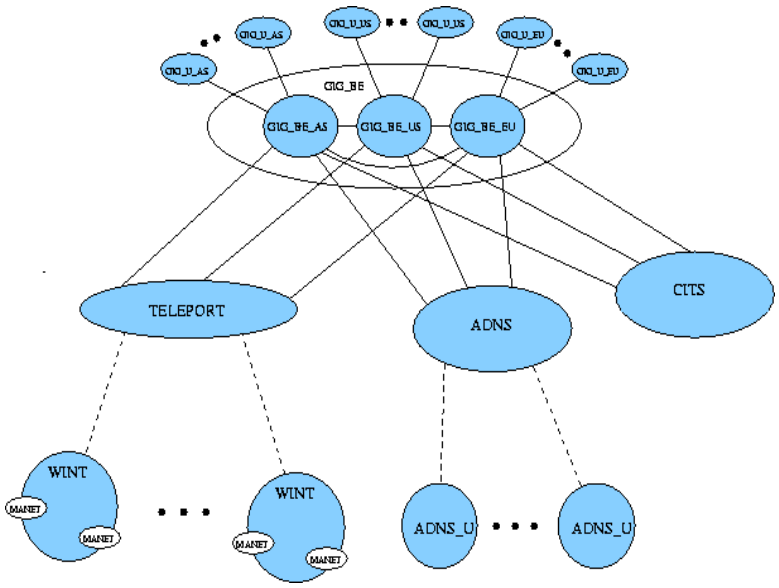

Figure 1: The high level topology of the near-term GIG model for simulation studies.

Additional GIG topology work is required to complete the characterization of the GIG for M&S purposes. This additional work includes characterizing the scale of the number of routing prefixes within the GIG for Intra-Domain routing purposes, characterizing the nature of network mobility, e.g., frequency of deployment of expeditionary forces from battleship groups to WIN-T Unit of Employments, and further specifics on the anticipated internal architecture of the various GIG networks identified in Figure 1.

6.3 NMS Tools Selection

In this section, we discuss the selection of tools developed within the DARPA NMS Program based upon their utility in supporting the performance studies we identified in Section 6.1.

6.3.1 Requirements

Our selection of NMS tools included the following criteria, which are driven by our set of identified performance studies:

- *Reusability* - a goal of our transition work is the development of a reusable GIG simulation tool of utility to the greater DoD community. This imposes requirements on the nature of the tools, their ease of configuration and inter-operation. It also implies that manual methods are not reasonable for the final, integrated simulation environment.

- *Applicability to Studies* - the studies require building a baseline GIG simulation supporting both wired and wireless links, e.g., satellite links, mobile ad hoc networks (MANETS), transport capabilities including Multi-Protocol Label Swapping (MPLS) and Differentiated Services (DiffSrv), inter and intra-domain routing protocols, and a vast range in link speeds ranging of OC-48 links down to low-speed wireless links on the order of 100 Kbps. Further, various application models and traffic models are required.

- *Scale* - Given the size of the GIG, parallel implementations of the simulation will be required. Also, some of the studies address vastly different time scales, e.g., investigate BGP performance over a high-speed network there packet transmissions are sub-microsecond while BGP time scales are on the order of minutes. This will require hybrid packet-level and flow-level modeling. At a minimum, the simulation tools need to support tens to hundreds of thousands of nodes. More likely, they will be required to simulate several million nodes.

- *Interoperability* - interoperability of the tools is extremely important for ease of use and for developing true end-to-end modeling. It is not feasible to build a reusable GIG simulation platform that requires manual methods for patching together separate simulations of various aspects of the network. Further, running separate, non-interacting simulations may neglect dynamic interactions between the various pieces of the GIG network being simulated separately.

- *DoD Open Source* - the tools used should be generally available to the DoD community. Tools developed under the DARPA NMS program fall within this category. For integration and further development, access to source code is required.

6.3.2 Simulation Tools

There is a diversity of excellent tools and capabilities developed within the NMS Program. However, there is no one single tool (or collection of tools) that would currently support the modeling of the diversity of topologies, technologies, and complexities found in the GIG and required to perform the above identified performance studies. In this section, we describe the selected set of NMS tools and capabilities, which we deem suitable for the execution of the above identified performance studies. We envision the development of a single GIG simulation

facility and selected NMS tools in this context. Further, we lay out a progression of tools and integration work necessary to incrementally build up a single GIG simulation necessary to execute the progressively more complex GIG performance studies.

We decided to build a single GIG simulation environment based upon the Network Simulation 2 (NS2)[16] simulation tool. When necessary, we would then migrate to the Parallel and Distributed Network Simulation (PDNS)[17] tool. NS2 was chosen as the base platform due to the broad support and breadth of contributions and capabilities made by the Research and Development community over the last fifteen years. NS2 has the broadest protocol support among existing simulation tools; including wired-based and wireless communications, extensive transport layer protocol support, IP and MPLS modeling, extensive Diff-Srv models, contributed routing protocol models for intra and inter-domain routing, application models, etc. PDNS represents an enhancement to NS2 for the express purpose of running simulation models in a distributed computing cluster. Both of these tools are available in open source and hence allow for code verification. In the NMS Program, the PDNS toolkit demonstrated a several million node simulation capability running on a distributed supercomputer at the Penn State University. Basing the GIG simulation facility on NS2 and PDNS affords a growth path to extremely large scale modeling of the GIG. Other tools and capabilities that we selected, are the BGP++ [4] code for modeling the BGP routing protocol, the Integrated Fluid Flow Models (IFFM) [14], the high fidelity Mobile Ad-Hoc Networking models incorporated into the GLOMOSIM tools [**?**], and the background traffic models based upon the work in [7] and [13]. As previously mentioned, these tools will require levels of additional integration coding in order to fully exploit their capabilities in GIG performance studies.

6.4 A Study Test Case

In this section we discuss our work in executing a test case. We felt it was necessary and useful to test our methodology against a case study. We chose, as an initial study, the use of the Border Gateway Protocol version 4 (BGP4) in the GIG, as discussed previously in Section 6.1. We investigated BGPv4 performance due to the fact that BGPv4 has proved to be a rather fragile protocol under certain situations within the Internet, combined with the fact that the GIG has several unique conditions and uses for BGPv4. These include a heavy reliance on relatively intermittent satellite links, a need for network mobility supported through BGP advertisements, and a need to continue operation under severe network failure conditions. Various types of satellite and terminal configurations are deployed or planned for the GIG, including fixed ground-based terminals, terminals mounted on mobile land-based vehicles, planes and ships. These satellite connections suffer channel fading due to obstructions, rain, antenna placement on ships and other mobile platforms. BGP4 must support routing over these intermittent links, requiring the proper protocol tuning and local decision policies. Further, it is important to investigate BGP4 traffic generation and rate of prefix propagation through the GIG under various assumptions regarding a) the mobility of networks and connections and b) the stability of the various satellite links.

We built the GIG simulation model for our test case on the NS2 and BGP++ tools. The NS2 and PDNS tools rely on the TCL scripting language to describe the simulation configuration and their core code is written in C++. BGP++ is written in C++ and relies on standard BGP daemon router configuration code for the simulation configuration. We developed a Practical Extraction and Report Language (PERL) script, to automatically generate the NS2 and BGP++

configuration code based upon a set of GIG input parameters, e.g., number of core nodes in the GIG-BE, etc. The PERL script generates configuration code for a broad set of parameter definitions and limits. The PERL code automatically generates the GIG models described above and allows for the definition of extremely large scale configurations.

We built a core hardware facility to host the GIG simulation model. Specifically, we deployed Apple Xserve Head and Cluster nodes for our distributed computing complex. These servers dual boot both MAC OSX and Yellow Dog HPC Linux. Distributed simulations will be supported with the Apple XGrid and the Linux Mosix distributed computing tools. The parallel network simulations will be supported via PDNS.

In our initial simulation studies, we ran a series of runs modeling a small three-node network configuration containing a single, wired and a single, intermittent satellite link. These small simulation studies served the following purposes:

- *Test and Verification* - it is important to perform test and Verification and Validation (V&V) on the various simulation tools under investigation. We performed an extensive set of runs to test and validate various aspects of the simulation tools. However, more effort is required in tools V&V.

- *Engineering Tools Development* - we used these simulation results to develop a engineering tool to estimate BGP protocol traffic overhead on the network. This tool can be used tune the BPG protocol in the presence of intermittent satellite channels. These results further demonstrate the value of NMS M&S tools in aiding GIG engineers and designers to achieve their objective of deploying a Global Information Grid.

We then consider the performance of the BGP4 routing protocol in a large scale simulation study modeling an expected architecture for the GIG. These larger scale simulation studies served the following purposes:

- *Test and Verification* - running our GIG simulation model and its associated scripts for topology generation, metric computation and report generation served to further test and verify the tools and our associated scripts.

- *Benchmarking* - we investigated the size limitations to running the GIG simulation model on a single threaded simulation environment, i.e., NS2, in order to understand when we would have to migrate to the PDNS simulation tool.

We first discuss the small, three node simulation studies over an intermittent satellite link. We then discuss the development of an engineering tool for BGP implementers derived from the simulation runs. Finally, we discuss our larger scale simulations of the GIG simulation model.

6.4.1 Simple Simulations for Test and Verification

We present a set of validation tests for the simulation tools and models before discussing the larger BGP simulation studies within a GIG environment. Figure 2 shows the configuration we use to perform test, verification, and model development. It consists of three BGPv4 routers sitting in three distinct Autonomous System (AS) domains. The two routers on the Left Hand Side (LHS) of the figure are peering over a T1 wired facility, while the two routers on the Right Hand Side (RHS) of the figure are peering over a 1.5Mbps satellite link.

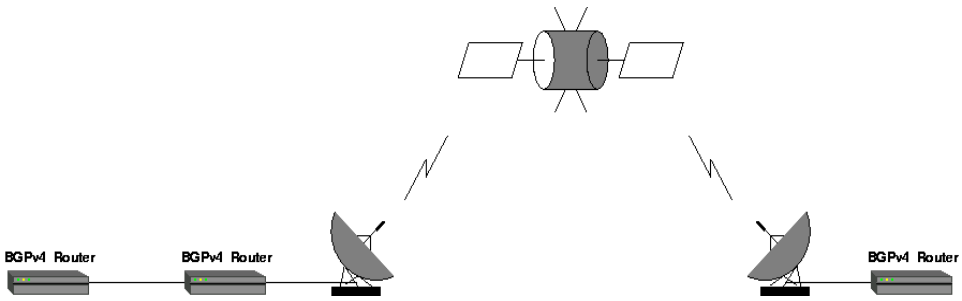

Figure 2: The BGPv4 test, validation and modeling reference satellite connection.

The baseline simulation platform for our initial BGPv4 stability studies is comprised of a) the Network Simulation 2 (NS2) event-driven simulation tool, b) the BGP++ module compiled into the NS2 simulation, c) the error link module from NS2, and d) our trace analysis and metric reporting scripts developed for this project. Our first set of runs were intended to test and verify the proper operation of these platform components. Our first runs are illustrated in Figure 3. The LHS plot uses the NS2 Error Model and the RHS plot uses the NS2 Run-Time capability to simulate the periodic satellite link. We used these two different and independent approaches within NS2 to simulate link dynamics as an additional V&V check of the model. Here we set up the BGP routers to advertise only 2 prefixes per router. We wrote a BGP log analysis script which parsed the BGP router logs and determined the transitions in the number of learned prefixes resident in the local router, not including the prefixes local to the router. We then set up a periodic satellite link using a) the Error Model and b) the Run-Time model within NS2 [16]. The figure shows the status of the link in the lower curve, i.e., 0 is down and 1 is up, and the number of learned prefixes in the upper curves for the router which is on the far LHS in Figure 2. Here the default values for the BGP Hold Down and Keep Alive timers were used, i.e., Hold Down timer = 180 seconds and Keep Alive timer = 60 seconds. For an excellent discussion on the operation of BGP, see [19].

Both plots show similar behavior. The period of the satellite link is 1000 seconds, i.e., it is down for 500 seconds then up for 500 seconds. The link down detection mechanism in BGP relies on receiving messages each Hold Down timer interval and forcing a message onto the link each Keep Alive timer interval, in the event that no other BGP traffic has been sent. Hence, the time for the BGP router to detect a down link is uniformly distributed between two and three Keep Alive times, i.e., between 120 and 180 seconds. This is demonstrated in the figures by observing that the time lag between the satellite link going down and the prefixes being withdrawn is between 120 and 180 seconds.

In Figure 4 we show the results of varying the Hold Down and Keep Alive timers on the time for the BGP peers to respond to the periodic satellite link transitions. For these runs, and as recommended in the BGP version 4 specification [2], we varied the Hold Down timer and set the Keep Alive timer to be one third the value of the Hold Down timer. The lower plot shows the response when the Hold Down timer is set to 18 seconds while the upper plot shows the results when setting the Hold Down timer to 270 seconds.

There is a concern that BGP traffic can cause a fair amount of congestion when links become intermittent and a large number of prefixes are to be advertised. When a BGP peering session is lost, all routes that were learned over the link are withdrawn from the network. Once the

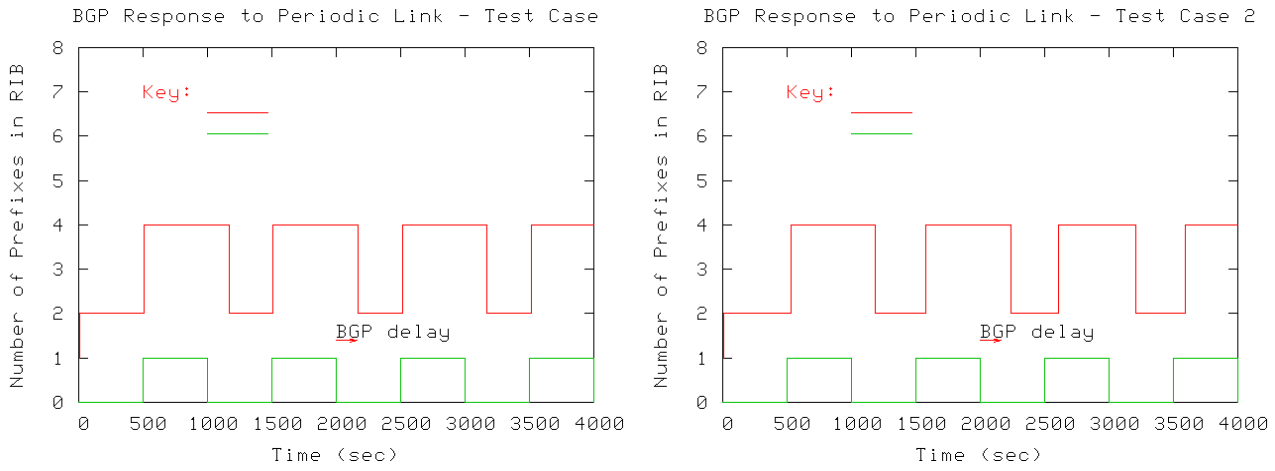

Figure 3: The BGP response to a simple periodic channel using two different NS2 mechanisms.

link is re-established, all prefixes in the BGP peers must be re-advertised over the new peering connection. Figure 5 shows the amount of traffic transmitted over the periodic satellite link as the number of prefixes to be advertised is increased. The traffic counts in the figure include traffic associated with the BGP peering over the link, prefix advertisements, Keep Alives, and all associated TCP overhead. It is apparent that the peak traffic, generated when the periodic link is re-established, increases linearly with the number of prefixes to be advertised.

In Figure 6 we show the results of varying the mean period of a random, intermittent satellite link on the traffic transmitted over the link. Here, the intermittent link behaves according to a Two State, Semi-Markov Gilbert-Elliot model [12] [11] as discussed in the next section. Each router is responsible for advertising 100 prefixes. On the plots, we indicate the mean period between transitions to the good, i.e., up, state. As the mean period decreases the amount of traffic increases over the link. We develop a model quantifying this effect in the next section to further validate the simulation results and to demonstrate value in running simulations of this type to build engineering tools for GIG design and deployment.

Based upon the above runs and others performed on this specific network configuration, we are confident that the NS2 Error Models and Run-Time models are behaving correctly. We are also confident that the BGP++ code, for the relatively simple configuration options tested, is behaving correctly.

6.4.2 BGPv4 Performance Modeling - BGP Overhead on an Intermittent Link

In this section, we discuss the issue of running exterior-BGP over a satellite link. The DoD places a heavy reliance upon satellite communications in order to reach remote nodes and networks. This reliance will continue in the deployment of the GIG. In fact, there is much discussion of running exterior-BGP over many of these satellite links in order to support dynamic topology changes due to network mobility, e.g., the deployment of a Marine expeditionary force from a battleship group into an Army Unit of Action (UoA) environment. The plan is to rely on BGP to advertise this small network domain change in connectivity across the GIG. However, there is little experience in running exterior-BGP over satellite links in the commercial Internet. For these reasons, we felt that demonstrating how the GIG simulation model could be used

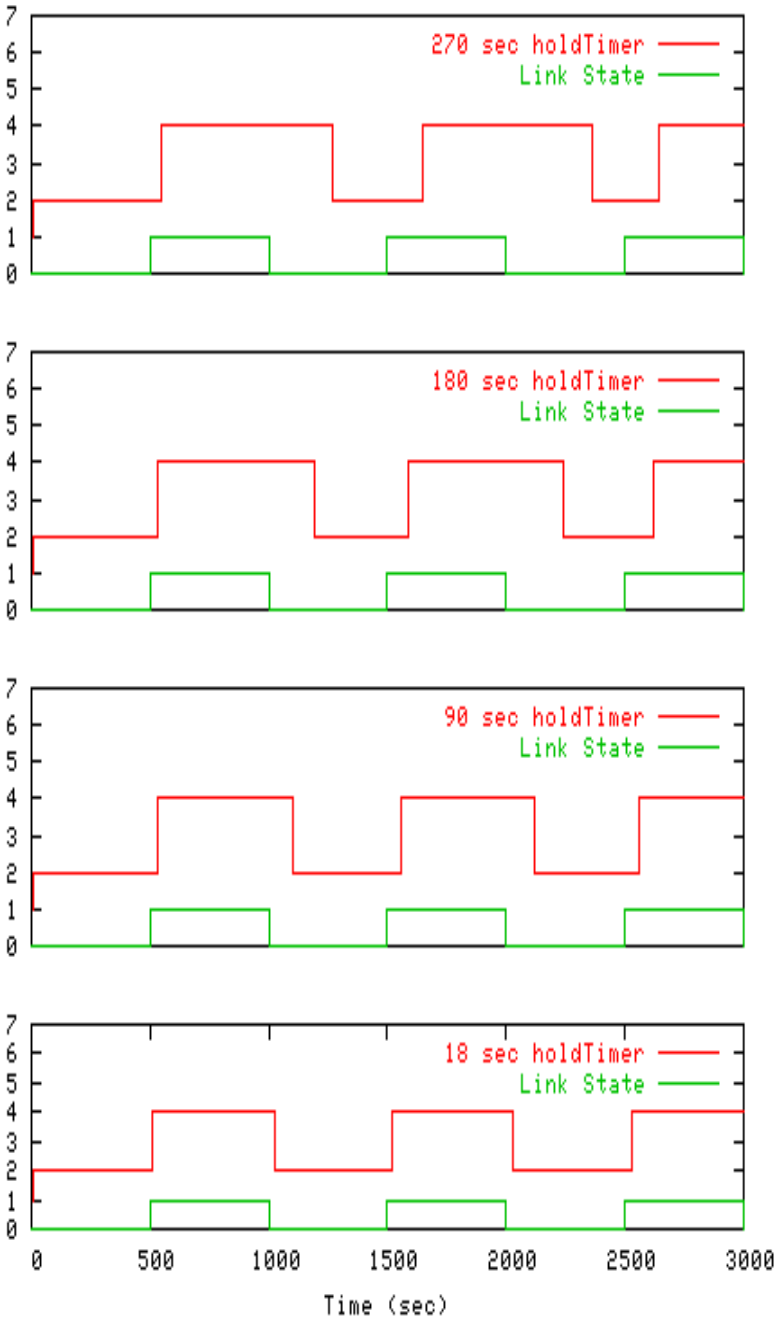

Figure 4: The BGP response to a simple periodic channel for different BGP Hold Down timer values.

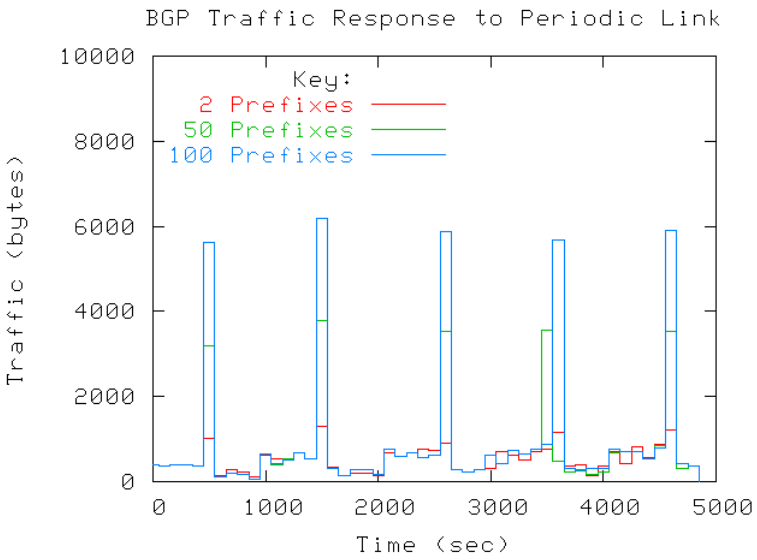

Figure 5: The BGP traffic transmitted over a periodic satellite link as a function of the number of prefixes advertised.

to quantify BGP traffic overhead in networks reliant on satellite links would encourage use of the simulation model in future DoD M&S studies. Further, developing an analytic model of BGP overhead and comparing with the GIG simulation model results would further validate the simulation model. Finally, an analytic model of BGP overhead over intermittent links could be a useful engineering tool for GIG designers.

We first consider a relatively simple Gilbert-Elliott model [12] [11] of the satellite channel. We then derive expressions for the traffic generated by BGP4 routing updates over this channel environment by treating the BGP4 state as the result of a Low Band-Pass Filter applied to the satellite channel state. The characteristics of the Low Band-Pass Filter are directly related to several BGP4 tuning parameters. We validate our traffic models by simulations of simple test case discussed in the previous section.

Figure 7 gives a pictorial representation of the Two-State, Semi-Markov, Gilbert-Elliot Model. This model is used to represent the satellite link being in one of two states, i.e., a good state (or UP-state) characterized by a low packet loss probability and a bad state (or DOWN-state) characterized by an extremely high probability of packet loss. In this model, we assume the system resides in a state for δ seconds before deciding to make a transition. The transition can be to the other state or it can be back to the same state. The transition probabilities are labeled p_{gg}, p_{gb}, p_{bb} and p_{bg} where p_{nm} represents the probability of transitioning from state n to state m. Only two of these transition probabilities are independent because of the relationships $p_{gg} + p_{gb} = 1$ and $p_{bb} + p_{bg} = 1$. Other more complex satellite channel models have been proposed, e.g., a three state Semi-Markov chain model. However, the Gilbert-Elliot model, which is a Two-State, Semi-Markov chain model, represents the simplest yet reasonably realistic model of satellite fading and is sufficient for initial studies.

When BGP is in its ACTIVE state [2], it is frequently attempting to establish a TCP connection to the far side exterior-BGP peer over the link. If the link is in the UP-state, the TCP connection setup and BGP handshakes will succeed and the BGP session will enter the

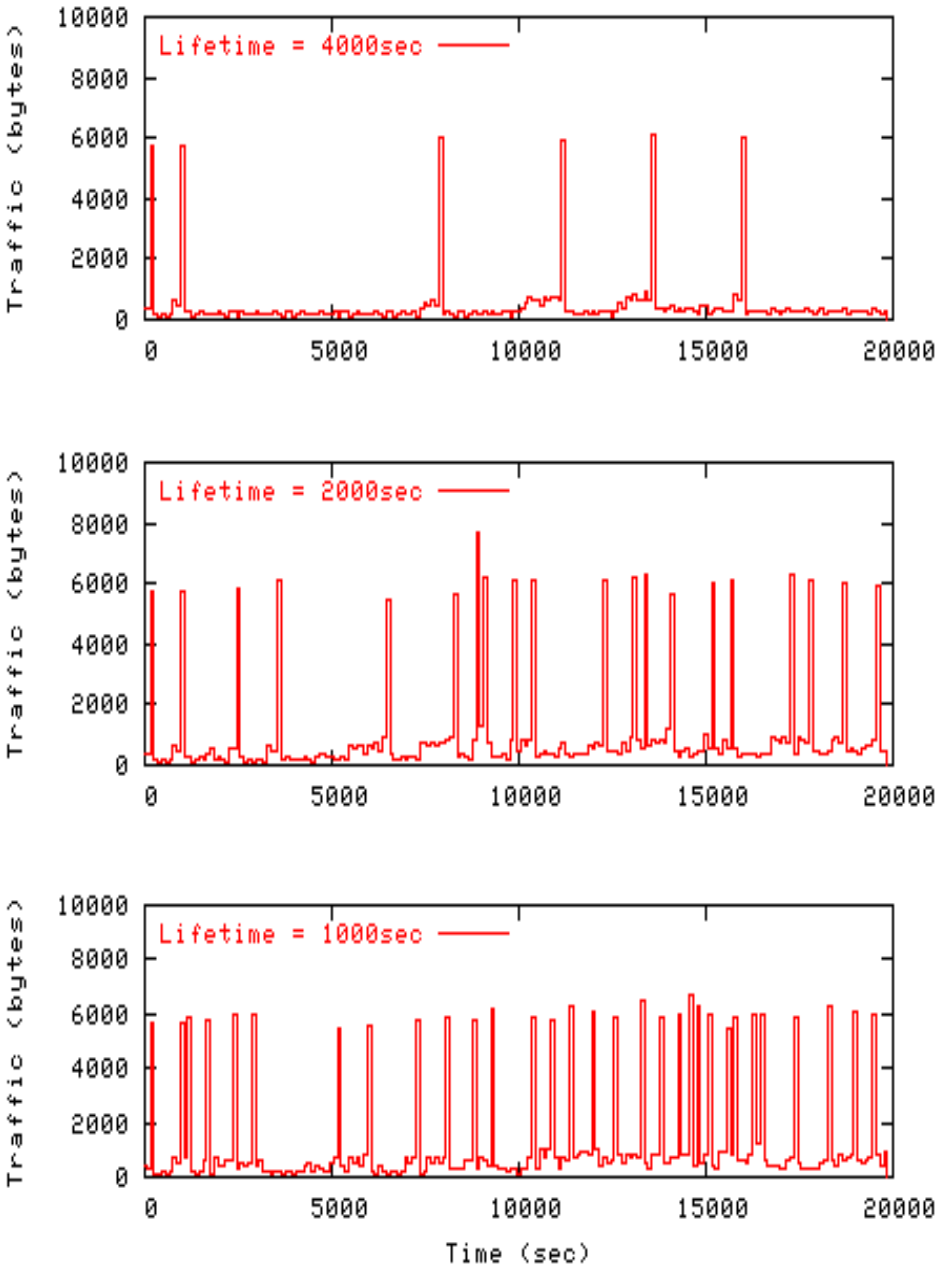

Figure 6: The BGP traffic transmitted over a random satellite link as a function of the mean cycle time of the channel model.

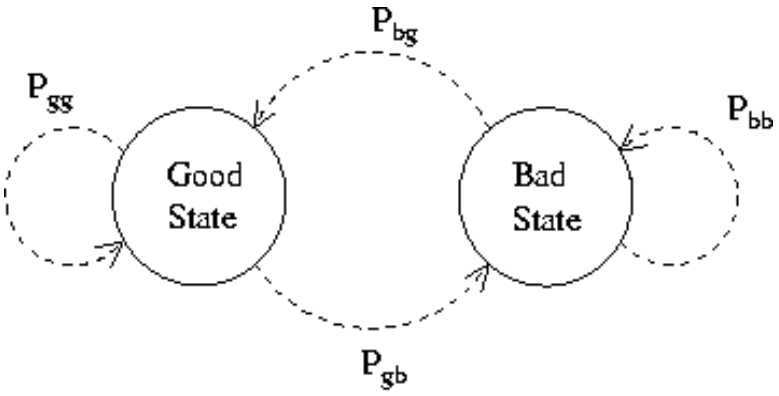

Figure 7: The two-state semi-Markov Gilbert Elliott model.

ESTABLISHED state. Once in this state, the BGP peers sample the connectivity to their peers each KEEP ALIVE seconds. They do this by setting a timer and ensuring that communications has occurred over the connection prior to the expiration of the timer. If the timer expires, then the BGP peers will send a KEEP ALIVE message to ensure that the link has been tested at least each KEEP ALIVE seconds. In the event that the KEEP ALIVE timer has expired N times in a row, where $N = HOLDDOWN/KEEPALIVE$, then the BGP peers declare the link down and the BGP peers enter the ACTIVE state again.

In order to model the BGP state transitions over the intermittent satellite link, we performed the following steps. We first developed a model of the satellite link, i.e., the satellite channel model. We then derived a model of BGP's perception of the underlying channel accounting for the BGP Keep Alive timer mechanism. We call this the "Sensed-State" model. Finally, we derived the BGP state transition model accounting for the impact of the BGP Hold Down timer mechanism. This model is derived by realizing that the Hold Down timer mechanism's effect is to apply a Low Band Pass Filter onto the "Sensed-State" model. These steps are represented in Figure 8.

We modeled this behavior as follows. We assumed that the BGP peers sample the state of the link each KEEP ALIVE time interval. BGP uses the status of these tests to decide whether it is to declare the link down or to continue holding it open. The underlying Gilbert model describes the actual state of the channel. The sampled channel state is the link state that BGP observes, so we first must transform the Gilbert model of the satellite channel to the 'Sensed-State' of the link as determined by BGP's KEEP ALIVE mechanism. Hence, we converted the Two-State, Semi-Markov model of the satellite link to a Two-State, Semi-Markov model of the sensed state. For example, if the KEEP ALIVE timer is set to $Mx\delta$ seconds where $M = 2$, then we must convert p_{gg} to p'_{gg} where p'_{gg} is the probability that the Gilbert model transitions from the good state to an arbitrary intermediate state and then to the good state in the second transition, i.e., $p'_{gg}(2) = p_{gg}p_{gg} + p_{gb}p_{bg}$ where the 2 indications the value of M in this example. In general, because the underlying Gilbert model is memoryless, we wrote these probabilities for an arbitrary $M > 1$ as:

$$p'_{gg}(M) = p_{gg}p'_{gg}(M-1) + p_{gb}p'_{bg}(M-1) \tag{1}$$

and

$$p'_{bb}(M) = p_{bb}p'_{bb}(M-1) + p_{bg}p'_{gb}(M-1) \tag{2}$$

Given that

$$p'_{gg}(M) + p'_{gb}(M) = p'_{bb}(M) + p'_{bg}(M) = 1 \tag{3}$$

and

$$p_{gg} + p_{gb} = p_{bb} + p_{bg} = 1 \tag{4}$$

we transformed these iterative equations to

$$p'_{gg}(M) = p_{gg}p'_{gg}(M-1) + (1-p_{gg})(1-p'_{bb}(M-1)) \tag{5}$$

$$p'_{bb}(M) = p_{bb}p'_{bb}(M-1) + (1-p_{bb})(1-p'_{gg}(M-1)) \tag{6}$$

where $p'_{gg}(1) = p_{gg}$ and $p'_{bb}(1) = p_{bb}$. Therefore, our "Sensed-State" Two-State, Semi-Markov Model is obtained from the Gilbert, Two-State, Semi-Markov Model of the satellite link by replacing the various unprimed transition probabilities with their corresponding primed counterparts obtained from the above iterative equation, and by replacing the step increment, δ, with a new step increment equal to $M\delta$.

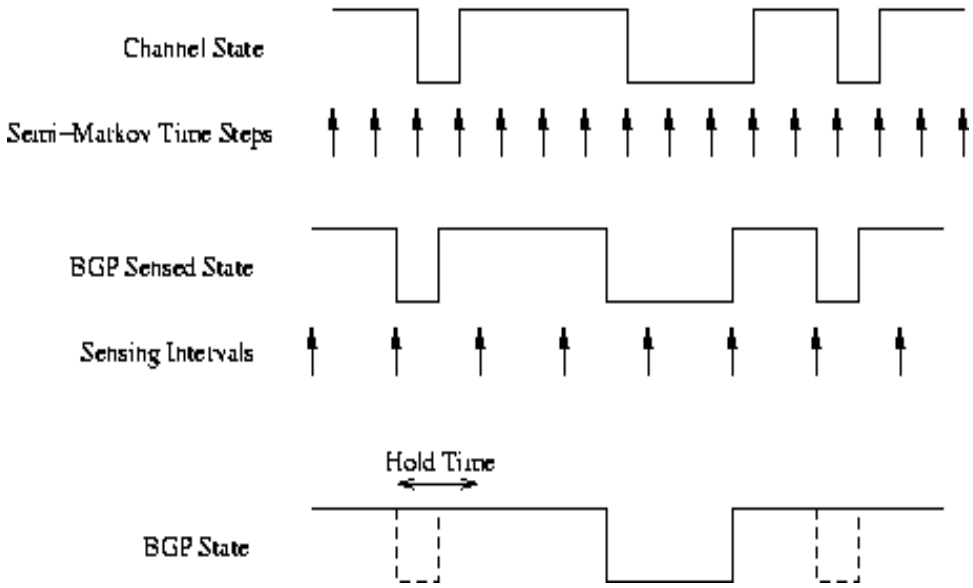

Figure 8: The three phases to build the BGP State Model starting from the top and working down, i.e., Channel State, Sensed Channel State, and BGP State.

Let us briefly discuss some interesting aspects of these two-state, Markov Models. An interesting quantity is the mean cycle time around the two states, i.e., the mean time for the system to enter the good state then transition to the bad state and then transition back to the good state. This mean cycle time can be written as the sum of the mean lifetimes in each state:

$$t^{(0)}_{cycle} = t^{(0)}_g + t^{(0)}_b \tag{7}$$

where $t^{(0)}_g$ is the mean time in the good state and $t^{(0)}_b$ is the mean time in the bad state. You can show that these times are related to the transition probabilities as:

$$t^{(0)}_g = p^{-1}_{gb} \tag{8}$$

$$t_b^{(0)} = p_{bg}^{-1} \tag{9}$$

where the same relations hold for the primed counterparts in the Sensed-State model as well.

So far we have effectively modeled the intermittent satellite link with a Gilbert Elliot Model and have modeled the sensing of the channel by the BGP KEEP ALIVE mechanism by our transformation to the Sensed-State Markov Model. BGP forms its view of the UP/DOWN status of the link based upon its configured value of the HOLD DOWN timer. This is typically set to an integer multiple of the KEEP ALIVE timer. So we assumed that the HOLD DOWN timer is set to $NM\delta$, where the BGP specification [2] recommends that $N = 3$. In the event that the BGP peer fails to receive any information over the BGP connection over a HOLD DOWN time, then it declares the link down and transitions back to the BGP ACTIVE state. The peer router then withdraws all routes learned over the (now) down link. Once the link is sensed to be up, the BGP peers re-establishes the BGP connection and advertise all paths over the recently up link. As long as the BGP connection remains open, the BGP peers only advertise changes in their Routing Information Bases (RIBs) (and not periodically re-advertise their entire RIBs). Due to the HOLD DOWN timer, the actual state of the channel can make relatively high frequency transitions, as long as the sensed state returns to up prior to the expiration of the HOLD DOWN timer. Therefore, the HOLD DOWN timer mechanism in BGP acts as a Low Band-pass Filter in deciding to transition the BGP state from ESTABLISHED to ACTIVE.

In order to model the impact of the HOLD DOWN timer, we determined the mean lifetime in the good and bad states as perceived by BGP. Clearly, for $N > 1$, the mean time in the perceived good state will be longer than $t_g^{(0)}$ because the HOLD DOWN timer mechanism will averaged out the cases where the channel had brief fades. We used the existence of renewal times to derive the perceived mean lifetime in the good state as follows: We demonstrate by first deriving the expression for the mean lifetime in a standard Two-State, Markov Model. The mean time in the good state (in units of δ) can be written

$$t_g^{(0)} = 1 + p_{gg} \times t_g^{(0)} + p_{gb} \times 0 \tag{10}$$

where the first term, unity, is due to the fact that we start our timer at the transition into the state and we are guaranteed of spending at least one time step in that state. The second term in the above expression addresses the case where the next transition is back into the good state times the remaining time in the good state. Because the system is memoryless, the remaining time in the good state is simple equal to the mean time in the good state. The final term in this expression covers the case where the next transition is to the bad state times the remaining time in the good state, which is zero. This expression can be inverted to find that $t_g^{(0)} = p_{gb}^{-1}$ as we had mentioned previously.

We extended this argument to the case where the HOLD DOWN timer mechanism is active, i.e., $N > 1$. In this case the question is whether the system can transition back to the good state before the expiration of the HOLD DOWN timer. Hence, we wrote:

$$t_g^{(N)} = 1 + p_{gg} t_g^{(N)} + p_{gb} \left[\sum_{i=1}^{N} Pr\{t_B = i\}(i + t_g^{(N)}) + Pr\{t_B > N\}(N) \right] \tag{11}$$

As before, the second term relies on the memoryless nature of the system and that transitioning back into the good state represents a renewal point. The third term accounts for the fact that even though the system transitions to the bad state it can still contribute to the mean lifetime in the good state via two mechanisms. The first being when the system transitions back to

the good state prior to the expiration of the HOLD DOWN timer times the length of time it took to transition back plus the mean time in the good state. The second mechanism accounts for the case where the system does not transition back into the good state in time, however it still contributes a time equal to the HOLD DOWN timer to the mean time in the good state because it takes the BGP peer that amount of time to declare the connection dead. Although it is rather tedious, this expression was inverted to yield:

$$t_g^{(N)} = t_g^{(0)} \left[\frac{1 + p_g b f^{(N)}(p_{bb})}{1 - h^{(N)}(p_{bb})} \right] \tag{12}$$

where:

$$f^{(N)}(x) = N(1-x)x^{(N+1)} + \frac{1 - (N-1)x^N + Nx^{(N+1)}}{(1-x)} \tag{13}$$

and:

$$h^{(N)} = 1 - x^N \tag{14}$$

We needed to address the mean perceived time in the bad state now. By the time BGP decides to declare the channel down, the channel has most likely be down for some time. So the question was what is the mean residual time perceived to be in the bad state. We wrote this as:

$$t_b^{(N)} = Pr\{ResidualLife | Life >= N\} \tag{15}$$

or:

$$t_b^{(N)} = \sum_{i=1}^{\infty} i Pr\{ResidualLife = i | Life >= N\} \tag{16}$$

Since the system is Markovian, this probability is simply the probability of the life equal to i, or:

$$t_b^{(N)} = p_{bg} \sum_{i=1}^{\infty} i p_{bb}^{i-1} = t_b^{(0)} \tag{17}$$

Therefore, the mean perceived life in the bad state is identical to the mean life in the bad state in the underlying Markov model.

Combining these expressions for the mean perceived lifetimes in the good and bad states, we got the mean BGP cycle time through the ESTABLISHED and ACTIVE states:

$$t_{cycle}^{(N)} = t_g^{(N)} + t_b^{(0)} \tag{18}$$

where $t_g^{(N)}$ is given in Eq.(12) above.

The Internet RFC1774 [3] gives an expression for the BGP overhead incurred due to prefix advertisements as:

$$T = O_H(P + HA) \tag{19}$$

where T is the network traffic generated by BGP prefix advertisements, O_H is the protocol overhead to communicate a path, P is the number of routes/prefixes advertised in the network, H is the mean number of AS hops per route advertised, and A is the total number of ASes in the network. For our intermittent link, this information must be transmitted each time the BGP state transitions to ESTABLISHED, which is proportional to the mean cycle frequency. Further, there is additional BGP overhead generated by the link maintenance functions which

Table 1: The simulation parameters testing the estimate of BGP overhead.

$t_g^{(0)}$ (sec)	p_{gg}	$t_b^{(0)}$ (sec)	p_{bb}	δ (sec)	M (sec)	N (sec)
250	0.8	250	0.8	50	30-360	90-1080
500	0.9	500	0.9	50	30-360	90-1080
1000	0.95	1000	0.95	50	30-360	90-1080

is proportional to the frequency of KEEP ALIVES. Thus, our expression for the BGP overhead on an intermittent satellite link is:

$$T = \frac{\alpha}{t_{KEEPALIVE}} + \frac{\beta}{t_{cycle}} \tag{20}$$

where α and β are considered as fitting parameters and $t_{KEEPALIVE}$ is equal to the Keep Alive timer value.

In order to test our expression estimating the BGP overhead over an intermittent satellite link we made a series of simulation runs. We simulated our simple three node network shown in Figure 2 above. We assumed each router represented a single AS advertising 1000 prefixes. The parameters for the Gilbert Model are shown in Table 1. Here we set the HOLD DOWN timer to equal three times the KEEP ALIVE timer, as recommended in [2]. We ran each combination of parameters 200 times for a simulated duration of 20,000 seconds. The traffic metric is the mean number of bytes per seconds transmitted over the satellite link averaged over the entire length of each simulation run. The fading satellite channel was modeled as the Gilbert-Elliot Two-State, Semi-Markov model where the transition parameters for the two states were identical and the mean lifetime in the good state was set to 250, 500, and 1000 seconds respectively.

Figure 9 shows the results of the simulation runs. We see, as expected, that the mean overhead traffic decreases as the HOLD DOWN timer is increased due to both the lessening of the effects of the KEEP ALIVE chatter and the prefix advertisements over the intermittent links. Also, as the mean lifetime of the good state is increased, the amount of overhead traffic decreases.

We performed a relatively simple fitting procedure to test our expression estimating the BGP overhead traffic, i.e., Eq.(20). We first set $\beta = 0$ and varied α until we fit reasonably well the maximum traffic point. This yielded $\alpha = 23,000$. We then set $\alpha = 0$ and varied β until we fit the maximum traffic point reasonably well. This yielded $\beta = 180,000$. These cases are shown in Figure 10. We see from these figures that the case where $\beta = 0$ does not make any distinction between the different satellite models. This is due to the fact that this case is solely dependent upon the value of the HOLD DOWN timer. Whereas the case of $\alpha = 0$ shows too flat a curve for the case where the mean lifetime of the satellite good state is large. This is due to the fact that this expression ignores the chatter over the link associated with BGP state maintenance.

We then took a weighted average of these two cases where the weightings are described by $\gamma\alpha$ and $(1-\gamma)\beta$ and found that $\gamma = 3/8$ gave a reasonable fit. This final case is shown in Figure 11. Where we see that this expression (with only two free parameters) does a reasonable job of characterizing the dependence of the BGP traffic overhead on the HOLD DOWN timer and the satellite channel model. This expression also shows a cross-over behavior between the

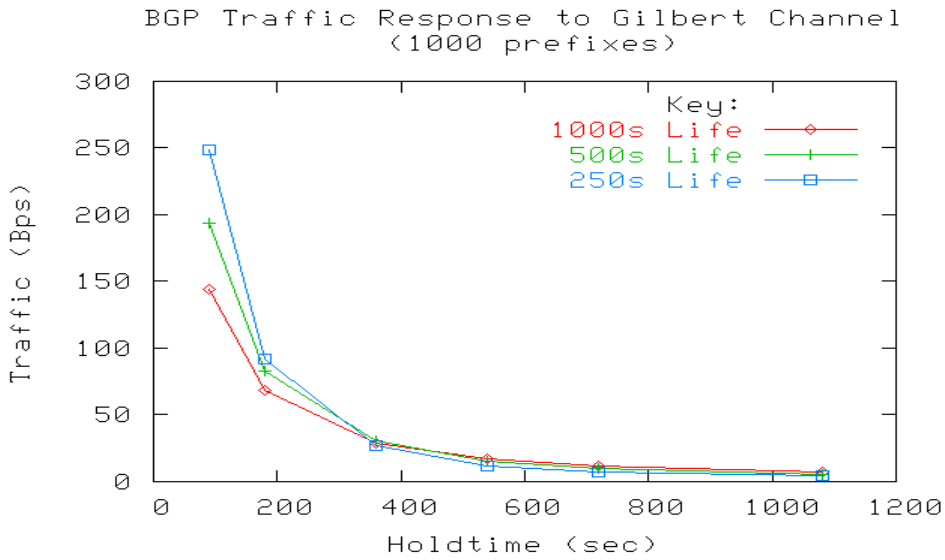

Figure 9: The simulation data of BGP traffic versus HOLD DOWN timer.

different satellite models as the HOLD DOWN timer is increased from 90 seconds up to 1080 seconds. We are encouraged by these results. But clearly more analysis is required before this type of modeling can be suggested for engineering guidelines in configuring BGP parameters for operation over satellite channels.

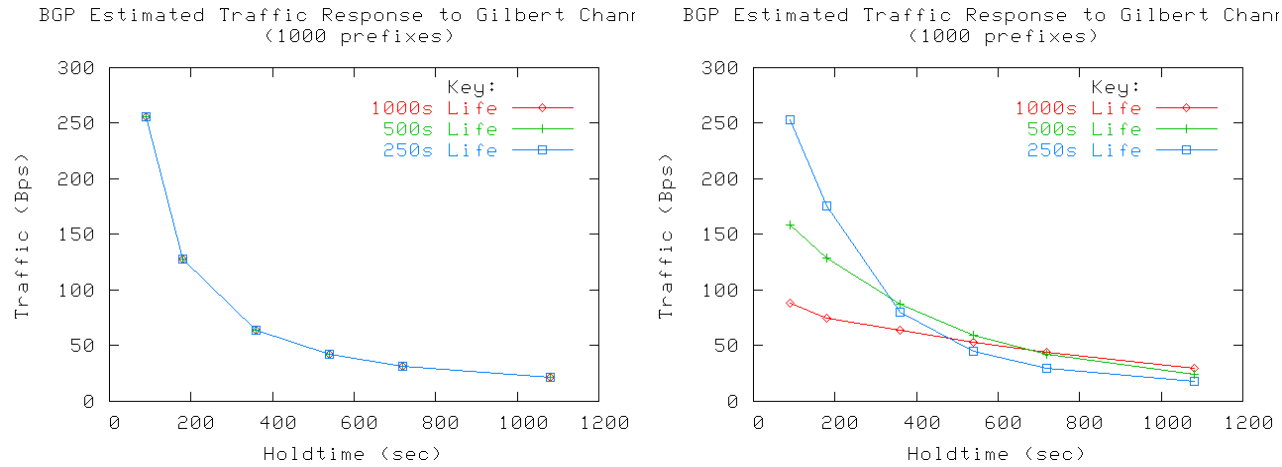

Figure 10: The fitting to the simulation data setting $\beta = 0$ (LHS) and $\alpha = 0$ (RHS) respectively.

Therefore, our model of the BGP overhead over intermittent links predicts well the results of the traffic overhead generated by the GIG simulation model. This serves to further validate the simulation model. We further believe that expressions of this type will prove of utility to GIG designers in tuning the BGP protocol for actual deployment. Finally, we believe that analytic results of this type, derived with the aid of the GIG simulation model built upon the NMS tools and capabilities will aid in further acceptance of these tools in the DoD M&S community.

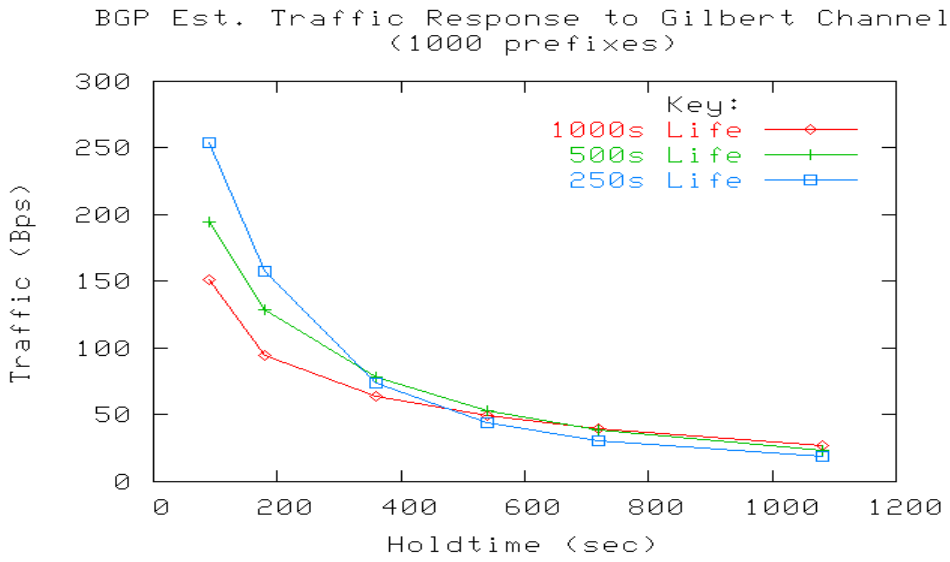

Figure 11: The fitting to the simulation data setting $\gamma = 3/8$.

6.4.3 Larger Scale BGPv4 Simulations

In this section, we report on our initial simulation studies of the larger GIG model we described in Section 6.2 above. Given the expected size of the GIG, once fully deployed, we anticipate having to support hundreds of thousands of routers in the simulation model. Further, the simulation platform should be flexible enough to provide an analysis of the GIG at various stages of its deployment. As such, we developed a PERL script, which automatically builds the simulation configurations for our studies.

As our test case focused on BGP modeling, the current configuration generation script builds the GIG topology, i.e., node definitions and interconnections, the dynamics of the satellite links, and the BGP ++ router configuration files. We also extended our BGP parsing scripts to handle these larger configurations and have modified the scripts to explicitly develop the satellite link dynamics instead of relying upon the NS2 Error Models. The simulation configuration script reads from an input file containing roughly 50 configuration parameters. The parameters include topology information, e.g., the number of ASes and routers within the GIG-Bandwidth Expansion network, capabilities, e.g., running BGP or not, and BGP router configurations for each BGP router in the model, e.g., the number of prefixes that each domains BGP routers are to advertise.

Our first set of tests of the GIG simulation consisted of GIG-BE, ADNS, GIG User, ADNS User, WIN-T, and a TELEPORT ASes. Each network domain was relatively modest in size; the entire network consisting of 50 BGP routers advertising a total of 80 prefixes. Our initial runs were to test and verify the operation of our configuration scripts, our metric reporting scripts and the NS2 and BGP++ simulation tools in a larger context. Figure 12 tracks the number of learned prefixes in a given BGP router's Routing Information Base (RIB) for each type of network domain. The plot covers both network initialization and the impact of a satellite failure at 300 seconds connecting to a WIN-T domain. The default values for the BGP timers are used for this set of results. It can be observed that it takes roughly 40 seconds for the BGP RIBs

to stabilize within the network upon startup. Further, it takes about 120 seconds for the BGP peers to declare the satellite link down and to withdraw routes from the network.

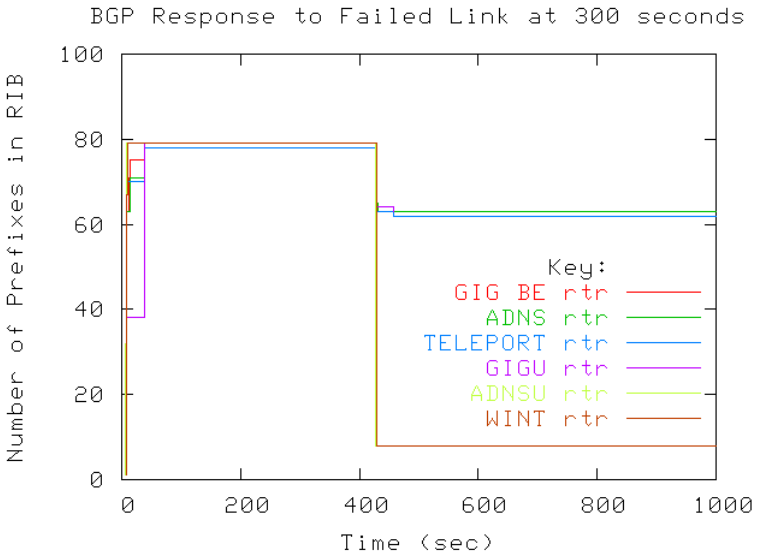

Figure 12: The BGP response in a 50 router implementation.

In Figure 13 we scaled the simulation roughly ten fold. Here, we simulated roughly 500 BGP routers advertising roughly 2000 prefixes in the presence of multiple intermittent satellite links connecting all of the ADNS User ASes, e.g., battle groups, and WIN-T ASes back into the GIG core network. Each satellite link is modeled via the Two-State Semi-Markov Gilbert-Elliot model with a mean time in the good state of 3000 seconds and a mean time in the bad state of 300 seconds. We ran the simulation for 20,000 simulated seconds; although we only show the results for the first 1500 seconds. The LHS plot shows an expanded view of the network startup, where it takes roughly 40 seconds for the RIBs to stabilize. Again, what we are plotting here is the number of prefixes learned by the router, not including those prefixes which are local to it. On the RHS plot we see that around 500 seconds into the simulation, an ADNS User AS loses connectivity to the GIG due to missing satellite connectivity.

Finally, a series of simulation runs were made on a single Apple G5 Xserve machine with 4 GBytes of memory and a 2.2 GHz processor. For this series of runs, the simulated time and the number of prefixes local to each BGP router were held fixed while the number of BGP routers was increased. In Table 2, we show the results for the memory consumption for each run. These values increase linearly with the number of routers within the simulation. Projecting to 90% utilization of the machine's 4 GBytes worth of memory, we estimated that a single-threaded simulation on this particular machine can support roughly 5200 BGP routers within the simulation. Obviously, to increase beyond this limit, we must move to using the PDNS simulation tool. Certainly on our ten node cluster, we could expect to simulate no more than 50,000 BGP routers. To exceed this limit would require migrating to a larger nodal cluster. We have already had some discussions to this effect (see Section 9).

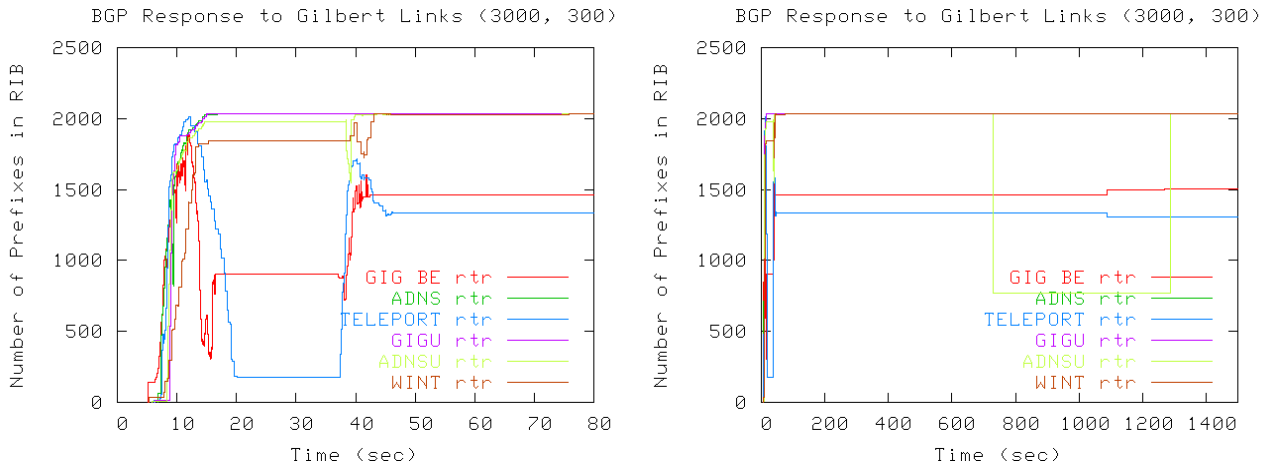

Figure 13: The BGP response in a 500 router implementation.

Table 2: The results from various simulation sizing runs.

Simulated Time (seconds)	Number BGP Routers	Memory Consumption (% 4 GBytes)	Run Time (seconds)
1000	44	0.6	109
1000	82	1.0	183
1000	152	2.4	1575
1000	347	5.7	-

7 IMPORTANT FINDINGS AND CONCLUSIONS

We summarize our important findings and conclusions in this section. They are:

- *Utility* - it is clear that NMS developed tools will be of use in the DoD's M&S community. We believe that the performance studies we describe are of value to the DoD. Further, our test case study of BGP performance in GIG-like environments demonstrated that the NMS Program has tools and capabilities which are immediately applicable to application in the DoD's M&S community.

- *Integration* - the NMS Program has produced an excellent collection of work and accomplishments, but interoperability did not seem to be a strong focus of the program. As such, we believe that there remains a fair amount of software development and integration work necessary to facilitate transition to the DoD. A set of recommended integration efforts are discussed in Section 10.1. These recommendations resulted from our execution of our test case study described in this report.

- *Transition by Design* - transition efforts should be included up front, as early as possible in the program life cycle, to influence technical design decisions to encourage acceptance by the DoD community. This may reduce follow on work to better integrate tools.

- *Tools Development* - many of the projects within the NMS Program resulted in associated software tools, but not all. A notable exception is the NMS work on Internet traffic characterization. This body of work could result in useful background traffic models for simulation tools.

- *Platform and Associated Support* - modeling tools require support tools for configuration, analysis, and reporting. Further, development on a common platform would aid in technology transfer. Much of work in executing the test case study involved the development of configuration, analysis and reporting scripts.

- *Follow On Recommendations* - see Section 10.

8 SIGNIFICANT HARDWARE/SOFTWARE DEVELOPMENT

As part of our efforts in demonstrating the value of the DARPA NMS Program capabilities, we executed a test case study of BGP performance over GIG-like environments. In the process, we had to develop a set of configuration, parsing and report generation scripts in order to execute our plan for technology transfer. We describe our software development efforts in this section.

We developed a configuration script to automate the generation of simulation configurations for the performance studies. The script was written in PERL, which offers extensive text manipulation, report generation and relatively simple syntax. Our PERL script currently builds the TCL-based configuration for the NS2 simulation package and generates the BGP router configuration files for the BGP++ simulation tool. The PERL script reads in a relatively simple test input file, which parameterizes the description of the GIG topology and the topologies of the component network domains. The input to the PERL script extends easily by adding new lines of the form {parameter_name parameter_value} to the input file. This variable is then automatically available for use within the PERL script.

The PERL-based, automatic simulation configuration script is roughly twelve hundred lines of code. It reads from an input file, currently containing roughly 50 configuration parameters. The parameters include topology information, e.g., the number of ASes and routers within the GIG-Bandwidth Expansion network, capabilities, e.g., running BGP or not, and BGP router configurations for each BGP router in the model, e.g., the number of prefixes that each domains BGP routers are to advertise.

By basing our configuration development in PERL, we have an extremely flexible capability which can be easily upgraded. For example, in the event that more general topology models of separate domains are required, we can easily incorporate these models into the PERL code. Further, PERL affords an excellent reporting capability. This could be exploited to automatically develop both the simulation TCL code for the simulation tools to digest, as well as human digestible reports describing the associated network configuration.

We developed several PERL scripts to a) parse the NS2 trace files in order to analyze for traffic patterns and plot generations, and b) to analyze GNU Zebra BGPD router log files for the current state of the router's RIBs for analysis and plot generation purposes. Example outputs from these scripts are represented in Figures 3, 4, 5, 6, 12 and 13.

9 SPECIAL COMMENTS

It was deemed important to socialize this work within the DoD community in order to stimulate interest to transition NMS Program capabilities to the DoD. In this section, we identify our efforts to do this as part of our DARPA NMS Transition activities.

- *DARPA NMS PI Meetings* - Dr. Bharat Doshi and Dr. Robert Cole attended the last two PI meetings. They used this opportunity to learn more about the projects within the NMS Program and to discuss the approach to working transition issues.

- *GIG Routing Working Group* - Dr. Antonio DeSimone from the JHU/APL is an active participant in the GIG Routing Working Group. This working group is addressing the preferred routing architecture within the GIG. The particular focus of their work is the design of the inter-domain routing architecture based upon the BGP4. In the January 2005 meeting of the GIG Routing Working Group in San Diego, CA, A. DeSimone gave a presentation on our efforts within the DARPA NMS program.

- *Army FCS Modeling Community in CERDEC* - Dr. Tibor Schonfeld from JHU/APL has participated in several meetings with the Army's Future Combat Systems (FCS) Modeling and Simulation Community. A purpose of these discussions was to socialize our DARPA NMS Transition Program work and to better understand the Army's needs in M&S.

- *Navy's Modeling and Simulation Technical Interchange Meeting* - R. Cole attended the 14th Technical Interchange Meeting (TIM) sponsored by the Navy's Office of Modeling and Simulation in March 2005 in Washington, DC. The purpose of the meeting was to better understand efforts by others to transition NMS activities into the DoD and to coordinate our efforts. We met with with representatives from the Navy's SPAWAR organization to discuss coordination of activities.

- *Army Research Laboratory/High Performance Computing Center* - R. Cole set up a meeting with the Chief of the High Performance Computing Division at the Army Research Laboratory at the Aberdeen Proving Ground, MD. The Chief of the High Performance Computing Division manages one of the four DoD High Performance Computing Centers as part of the DoD High Performance Computing Modernization Program (HPCMP). The purpose of the meeting was to present our work on building a GIG Simulation platform, to investigate interest opportunities at co-development to extend the NMS capabilities and to explore computing opportunities at the DoD's HPCMP.

- *DoD's High Performance Computing Modernization Program* - R. Cole is in the process of setting up a meeting at the DoD's HPCMP office in Washington, DC to explore interest in future GIG simulation platform development based upon NMS tools and capabilities as identified in Section 10.

- *Navy SPAWAR* - We are in the process of setting up further discussions with SPAWAR out in San Diego, CA.

- *External Publications* - We have written and submitted a series of papers to various conferences in order to further the socialization process. Specifically, we have submitted two abstracts to MILCOM 2005 on our efforts to build a reusable simulation platform of the

GIG based upon the NMS tool set [9] and our performance studies to investigate QoS enhancements to BGPv4 for the GIG [1]. Further, we were invited to present the results of our BGP4 Test Case Study at the IEEE PacRim Conference on Communications in August 2005 [10].

10 IMPLICATIONS FOR FURTHER RESEARCH

Much work is necessary to achieve the goal of developing a reusable GIG simulation model based upon the DARPA NMS tools and capabilities for use within the DoD community. In this section, we identify several key work items.

The following tools enhancements were identified and are desirable:

- *Verification and Validation (V&V)* - extensive V&V is necessary to test the simulation tool sets at each stage of the development. A possibility is to leverage the work of the GIG-Evaluation Facility (GIG-EF) Working Group to accomplish this. Another, less desirable option, is to rely on extensive and simple test runs which can be validated by hand.

- *Application Traffic Models* - there is a need to develop a set of DoD application models and turn these into useful, integrated traffic generation tools for the simulation studies of the GIG.

- *Improved GIG Topology Models* - work to continue to refine and improve the GIG topology models for the simulation platform is necessary. Our work to date [8] did little to address the networking plans related to the U.S Air Force and its relation to the GIG deployment. Another area of future work is incorporating a better, more detailed view of the intra and inter-domain routing architectures, once these are developed by the relevant GIG Working Groups.

- *BGP Routing Integration* - currently the BGP++ simulation tool, developed at Georgia Tech University, models only the control messages passed between the BGP routers and the BGP processing within each router. It does not interface with the routing modules within the NS2 and PDNS simulation packages.

- *BGP QoS Enhancements* - BGP currently does not support QoS routing; hence the BGP++ tool does not support QoS routing. We plan on enhancing the BGP protocol to support QoS routing and to simulate these extensions in the BGP++ tool.

- *Integrated Fluid Flow Models* - the IFFM models, developed at UMASS-Amherst, do not currently coordinate the flow path with dynamic routing decisions in the NS2 simulation tool. In order to investigate the impact of QoS routing capabilities in BGP, it is necessary to measure/simulate the impact of these enhanced routing decisions on the overall end-to-end packet flows across the GIG network. This will require integration of the IFFM with the dynamic routing within the NS2 tool.

- *Integrated Background Traffic Models* - there were several work projects within the NMS program investigating the characteristics of Internet traffic. These models would be useful in the development of background traffic models for simulation platforms such as the GIG simulation platform. This would require taking the results from these efforts and developing a background traffic module in NS2.

- *Improved Simulation of Wireless Physical Layer* - although we have not discussed this in much detail above, the work at UCLA to develop high-fidelity, physical layer wireless models into their simulation is extremely valuable. These models need to be ported to the NS2 simulation tool in order to enhance the GIG simulation model.

11 ACKNOWLEDGMENTS

The authors gratefully acknowledge numerous and useful discussions with Antonio DeSimone, Lotfi Benmohamed, Phil Chimento and Kan Schmidt of the JHU Applied Physics Laboratory. We wish to thank Dawn Tolle and Edu Camarah for their support in the management of this project. Further, we wish to thank Kathy Raetz, Roger Hammons and Eric Naber our their extensive review of this report. Finally, we wish to thank Xenofantos Dimitropoulous from Georgia Tech University for kindly answering our questions on the configuration of the BGP++ tool.

References

[1] Benmohamed, B. Doshi, DeSimone, A. and R.G. Cole, *QoS Enhancements to BGP in the Global Information Grid*, abstract submitted to MILCOM05, February 2005.

[2] Rekhter, Y. and T. Li, *The Border Gateway Protocol - 4 (BGP4)*, Internet Engineering Task Force (IETF), RFC 1777, November 1995.

[3] Traina, B., *The Border Gateway Protocol - 4 Analysis*, Internet Engineering Task Force (IETF), RFC 1774, November 1995.

[4] The BGP++ simulation, *http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/*, February 2005.

[5] Dimitropoulous, X. and G. Riley, *Simulation Tool for Border Gateway Protocol Studies*, Internal Georgia Tech University Report, November 2004.

[6] Quoitin, B., Pelsser, C., Bonaventure, O. and S. Uhlig, *A performance evaluation of BGP-based traffic engineering*, preprint, December 2004.

[7] Cleveland, W., Purdue University, *Internet Traffic Models*, http://www.stat.purdue.edu/∼wsc/, February 2005.

[8] Cole, R.G., Benmohamed, L., Chimento, P., DeSimone, A. and B. Doshi, *A Topology Model of the Global Information Grid for M&S*, JHU/APL Internal Technical Report, February 2005.

[9] Cole, R.G., Benmohamed, L., DeSimone, A. and B. Doshi, *Initial Investigations of BGP Performance on a Global Information Grid Simulation Platform*, abstract submitted to MILCOM05, May 2005.

[10] Cole, R.G., Benmohamed, L., DeSimone, A. and B. Doshi, *Border Gateway Protocol 4 (BGP4) Performance over Intermittent Satellite Links*, invited talk at IEEE PacRim Conference on Communications, August 2005.

[11] Elliot, E.O., *Estimates of error rates for codes on burst-noise channels*, Bell Syst. Tech. J., vol.42, pp 1977-1977, September 1963.

[12] Gilbert, E.N., *Capacity of a burst-noise channel*, Bell Syst. Tech. J., vol.39, pp 1253-1265, September 1960.

[13] Riedi, R., Rice University, *Internet Traffic Models*, http://www.stat.rice.edu/∼riedi/ February 2005.

[14] Towsley, D., University of Massachusetts - Amherst, *Integrated Fluid Flow Models*, http://www-net.cs.umass.edu/, January 2005.

[15] Nichols, D. et al., *SSFNet*, http://www.ssfnet.org/, February 2005.

[16] The Network Simulator 2, *http://www.isi.edu/ns2/nam/*, March 2003.

[17] Fujimote, R., Georgia Tech University, *The Parallel and Distributed NS2*, http://www.cc.gatech.edu/compass/pdns/index.html, January 2005.

[18] Ekici, E. and C. Chen, *BGP-S: A Protocol for Terrestrial and Satellite Network Integration in Network Layer*, in Wireless Networks, vol. 10, pp. 595-605, 2004.

[19] Stewart, J., *BGP: Inter-Domain Routing in the Internet*, Addison-Wesley Press, Boston, MA, 2001.

[20] GNU Zebra, *The GNU Zebra Routing Demon - Border Gateway Protocol*, http://www.gnuzebra.org/, January 2005.