AFIT/GCS/ENG/97D-12

A MODELING AND SIMULATION APPROACH TO CHARACTERIZE NETWORK LAYER INTERNET SURVIVABILITY

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Systems

Leif S. King, B.S.

Captain, USAF

December 1997

AFIT/GCS/ENG/97D-12

A MODELING AND SIMULATION APPROACH TO CHARACTERIZE NETWORK LAYER INTERNET SURVIVABILITY

THESIS

Leif S. King, Captain, USAF

AFIT/GCS/ENG/97D-12

The views expressed in this thesis are those of the author and do not necessarily reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/GCS/ENG/97D-12

A MODELING AND SIMULATION APPROACH TO CHARACTERIZE NETWORK LAYER INTERNET SURVIVABILITY

THESIS

Leif S. King, B.S.

Captain, USAF

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Systems

Jack M. Kloeber, LTC, USA Member	Rick A. Raines, Maj, USAF Member

Gregg H Gunsch

Gregg H. Gunsch, Lt Col, USAF
Chairman

Acknowledgments

This thesis would have never been possible without the guidance of my advisor Lt Col Gregg Gunsch. His tutelage provided focus and saved many hours of what would have been less productive work. His approach to the research fostered the natural curiosity inherent in the learning process and made the task less daunting and more interesting. I would also like to thank Ed Caine and Craig Labovitz whose knowledge and insight into the research area were beneficial to this effort.

The main reason that this thesis was possible was the unwavering support of my family. From my daughter Aimee's help in manually filling out a 2500 member matrix (I used to think that 2500 was a small number), to the gentle and confident support provided by my youngest daughter Abigail, to the empathetic and unceasing support provided by my wife, Susan, who was a positive role model during this effort, I couldn't have gone wrong. The sense of accomplishment felt at this moment is only surpassed by my love and dedication to them.

Leif S. King

Table of Contents

Page

List of Figures

Page

List of Tables

Abstract

The Air Force Core Competency of Information Superiority will be achieved in an age of decreasing AF manpower and corporate expertise. Increased AF reliance on COTS solutions, coupled with nearly ubiquitous points of entry to communication networks, create unique challenges in maintaining the Information Superiority edge.

The protection of the internet is part of this equation. The internet supports the daily business traffic of the Air Force. Personnel, finance, and supply data flow through its routers. Controlling an adversary's access to our information systems, either the data, or the hardware and software that control the data and transform it into information, is a key operation of Defensive Information Warfare which is the primary focus in maintaining Information Superiority.

This research will attempt to answer the viability of implementing measures designed to ensure the survivability of the internet communications infrastructure against Denial of Service attacks. It will provide planners the information to make decisions based on the cost and benefit tradeoffs associated with such measures. The requirements of system survivability are a superset of those that ensure security. The Air Force will need the cooperation of outside agencies to build survivability into the systems we rely on, but don't necessarily control.

A MODELING AND SIMULATION APPROACH TO CHARACTERIZE NETWORK LAYER INTERNET SURVIVABILITY

I. Introduction

Motivation

With the dissolve of the former Soviet Union the ideas of Clausewitzian warfare have been replaced in the Air Force by doctrine stemming from first Global Reach-Global Power, and most recently from Global Engagement: A Vision for the 21st Century Air Force [5]. As part of the National Security Strategy these visions represent the anticipated operation of the Air Force in a world where our enemies are less well known, their challenges to us are less predictable, and our expected range of response may be more varied and less nation-state oriented. Add to this mix a global technological proliferation occasioned by the rapid spread of information made possible by unparalleled growth in the internet, ubiquitous mobile communications available to almost anyone almost anywhere, direct television reception (and with a little more know-how and capital, transmission), and the changing role of information in warfare from an adjunct to weapons to a weapon/target in its own right, and we have the justification for the Information Superiority Core Competency [5:1-14].

Research Focus

Information Superiority relies on secure, robust, and survivable information systems. This research will focus on the survivability aspects of the internet in the presence of Denial of Service (DoS) attacks against the infrastructure. A DoS attack seeks to deny service to the target of the attack. It is not so much concerned with targeting specific information, but with taking advantage of holes in operating systems and communications protocols to disallow access to any information on the target system/group of systems. Traditional security measures to protect systems use mechanisms such as passwords and firewalls that are designed to keep out intruders. These measures may not be an effective deterrent to an attacker initiating a broader-swath DoS attack because the target system of the DoS attack can be outside of the firewall. Mere protection from without (the traditional security approach) is not sufficient if a DoS attack can cut off communications from a point outside the firewall.

The research is timely because the DoD, like many other businesses, is taking advantage of the economies of scale offered by internet use, but little has been done to ensure that the medium is survivable outside the original design considerations. The internet carries data that represent the day to day business of the AF: messages, deployment data, supply data, finance transactions, to mention a few. The long term compromise of this business could compromise the C^2 of an active campaign [15:9]. The more reliant the DoD becomes on the commercial infrastructure, the more long-term an outage caused by a DoS attack could become. Also, this research is directly extensible to other Internet Protocol (IP) routed networks such as Secret IP Routed Network (SIPRNet), which carries Global Command and Control System (GCCS) data. This is because its data courses the commercial internet backbones (albeit safely encrypted). Unlike unclassified DoD data traffic, GCCS data has the further protection of only being switched within the SIPRNet routers. To negatively impact GCCS traffic, the DoS attack would have to be extensive enough to force extremely high traffic rates on the shared communications backbones. In consideration of unclassified traffic, both the internet routers and transmission media are points of attack.

Systems survivability represents more than measures designed to repel specific forms of attack. The term survivable system refers to systems whose components collectively accomplish their mission even under attack and despite active intrusions that effectively disrupt some significant portion of the system [16:3]. This refers to the routers and protocols used to protect and route data such as the internet protocol mentioned above. It also refers to the design of the network to include resiliency factors such as path redundancy, node degree, and logical hierarchy.

Background

With increasing Air Force reliance on commercial-off-the-shelf (COTS) solutions for its information technology needs and downsizing of the force representing loss of expertise in systems security areas, the AF is increasingly vulnerable. The present trend toward increased sharing of common infrastructural components in the interests of economy will ensure that the civilian networked information infrastructure will always be an inseparable part of our national defense [16:2]. Sharing the common components has benefits of standardization, extensibility, and reusability. We also save money in R&D and maintenance budgets, but it has its drawbacks too. First, because of the nature of our business, commercial companies may not motivated by the same stringent security requirements as is the DoD, so the design of their products may make them a rough fit to an AF application. Second, commercial information systems products are well known and their bugs are widely exploitable. This is why the Computer Emergency Response Team (CERT) organization, for one, has an area reserved for vendor submissions where vendor specific bugs get the same distribution as do open systems bugs.

Prior to the Air Force use of the internet, data traffic that used to be part of hybrid systems was protected by the nature of the system. Adversarial targeting of varied and dissimilar systems whose data is less attractive to compromise than the compromise of an aggregation of data from many systems, as exists on the backbones of the internet, provides at least some measure of protection. Also the vulnerabilities of the internet are well understood and widely exploitable. With the DoD move to leverage the commercial communications infrastructure, more of our eggs are in one basket.

The concern over computer systems vulnerabilities caused at least one well known company of national scale to lock up its firewalls so tightly as to disallow most communication from the outside world to include turning off the mail port [18]. This can have the same effect as a DoS attack. The point is, that designing a system to be so airtight as to keep out all imaginable types of attacks shuts out needed inter-network communication between "friendly" entities and is an idea somewhat orthogonal to communication itself.

Apart from the initial specification, very little research has been done to date to incorporate survivability measures in the internet. Addressing this issue is a large task because of the way the internet has evolved and the way it is growing today. The present day internet is an ad-hoc mixture of components grown out of the ARPANET. Like any evolving system, legacy constructs necessitate the design of backward compatibility that leaves holes in the design structure. These holes are exploitable from a security and survivability standpoint.

Research Problem

This research will attempt to answer the viability of implementing measures designed to ensure the integrity of the internet communications infrastructure against DoS attacks. It will focus protecting the traffic designed to maintain the veracity of the infrastructure in its operational mode. This traffic is the control traffic on the internet. This control traffic establishes and maintains a current picture of the internet and its ability to deliver data from source to destination. This picture may change as portions of the internet become more or less busy. The control traffic ensures that these changes are communicated to the switches on the internet so that these switches operate with the latest information.

Over the last few years, as the internet has grown in popularity, the amount of traffic it carries is threatening the correct operation of the control traffic. That is, portions of the internet from small isolated sections, to regional and even national sections, have begun to experience natural DoS conditions on a daily basis as the control traffic begins to behave incorrectly.

The employment of specific mechanisms to ensure survivability in the presence of intentional DoS attacks will add extra processing overhead to the infrastructure and perhaps exacerbate the conditions leading to the natural DoS states. Therefore it is essential to characterize the effects that introducing these mechanisms will have on internet operation and attempt to answer questions associated with risk assessment such as, "Will the added overhead be acceptable considering the potential harm caused the internet infrastructure by launching DoS attacks targeting its control traffic?" Modeling and simulation information will be presented to illustrate the feasibility of this. From this study, specific recommendations can be made that will provide information on what mechanisms to use to effect internet survivability and how to deploy them.

Scope

This research will focus on the survivability issues in the internet motivated by the perceived threat that DoS attacks carry and tempered by the state of the normal operational mode of the internet that is exhibiting natural DoS conditions as it becomes more saturated. In the context of internet survivability, certain traditional security measures such as encryption, hashes, and digital signatures are considered. From a survivability perspective however, it is not that the particular tools to ensure survivability are different from those employed to provide security, it is that the requirements of a survivable system mean that these common tools may be combined and used in unique ways.

Most work has been done to date on security inside an internet domain and has been host specific. The focus has been on systems where there is centralized administrative control for security issues and controlled isolation from other internetwork entities. Firewalls, and to a large extent, the advisories posted by the CERT, are applicable to the security paradigm. Instead of considering application, session, or transport level protection measures, mechanisms that are more commonly associated with data security than survivability, the context of this study will be at level three of the OSI communications model. It concentrates on hardware and software at the network level. The links that will be studied are outside any particular domain boundaries. That is, outside the logical area of administrative control for any one network administrator. The scope is further constrained by considering DoS attacks from the standpoint of the *control* traffic on the internet. That is, the protocols that control the switch (or routing) function. A potential attacker could hope to have far more devastating effects by manipulating internet control traffic than instigating DoS attacks against particular host systems. An attack against a particular host will presumably take that host offline but the attack against infrastructure control can take many more hosts off line and for longer periods of time. The distribution of an attack of this nature is much larger in scale. In this context, where the communications infrastructure is concerned, the issues of security become issues of survivability.

II. Literature Review

Introduction

Sections in this chapter include a discussion of system survivability and why it is of concern to Air Force internet traffic. Next it considers studies about survivability characteristics of an analogous system: the Public Switched Telephone Network (PSTN), and why the PSTN analogy may become more applicable to the internet as the internet grows and new types of service are being demanded. Next it looks at system survivability in the context of DoS threats to the internet. Sample DoS attack scenarios are presented, then requirements are specified that are intended to thwart/mitigate such attacks. Chapter 5 presents data explaining the effects of implementing those requirements at the internet routing level.

System Survivability

System survivability is the capacity of a system to complete its mission in a timely manner, even if significant portions of the system are incapacitated by attack or accident [16:1]. Survivable system concepts include the disciplines of software engineering and computer science such as reliability, fault tolerance, verification of correctness, and security. But the practices associated with survivability are still being defined as are the actual specifications and requirements. The survivability paradigm is much broader than the security paradigm. Current security mechanisms are threat specific, narrow in scope, and are not effective in detecting an attack, recovering from an attack, or helping a system to survive a breach and complete is mission in spite of incongruent or malicious activity. Furthermore, security mechanisms are a patchwork of after the fact actions aimed at shoring up systems that had little or no consideration for the design of security (much less survivability) services in their specifications. A recurring theme found in studies of information systems is that security, ergo survivability, is not considered in the design phase.

An example of a security mechanism is a public information system such as the CERT. The CERT is a clearinghouse for open systems and vendor specific security vulnerabilities. It solicits information on, and verifies security bugs in operating systems and applications, and distributes the information to systems administrators. But the CERT Bulletins are published mostly after reported breaches in security. Indeed, the formation of the CERT itself was an after-the-fact response to the famous Morris Internet Worm attack. As with many other systems, the realization that security measures are lacking is realized only after breaches in security.

In contrast to the application of security measures, early work in creating survivable systems is focusing on detailed specifications and plausible requirements. In the case of resiliency to DoS attacks on the internet, one Defense Advanced Research Projects Agency (DARPA) sponsored study characterizes their requirements generation this way:

> The approach adopted here is primarily top-down, driven by the notion of correct operation of the [internet] protocols. However, the granularity of the requirements is influenced by knowledge of attack characteristics and knowledge of security countermeasures characteristics. The goal is a requirements characterization that introduces an appropriate level of specification granularity to reflect the implications of various types of attacks that might be mounted against a routing system while considering the costs of employing various mechanisms to detect and/or counter these attacks. [10:7]

This type of approach is proactive because the methods to meet a survivability requirement are considered in the light of the known and plausible methods of affecting a DoS attack on the internet.

A useful paradigm to focus the research methods would be to cast survivable systems as a sociological analog to public health efforts. These efforts are designed to prevent a broad spectrum of illnesses and maladies that face a community through such measures as immunizations, general cleanliness, shelters, and so on. The public health effort is generally meant to be proactive in nature and its focus is not on any specific problem, but it affects solutions to mediate the general health. Last ditch emergency efforts involve fighting specific maladies through measures such as quarantine or other specific treatments but this usually assumes a failure of one part of the system to do its job or a disaster on a large enough scale to affect an overload of the system.

In the case of the internet, detailed specification and requirements work can be done based on new technologies, but effectiveness may be hampered by interoperability and compatibility issues because security concerns were not a primary consideration in the design of the ARPANet. Ironically however, survivability was, as the ARPANet was originally designed to provide communications capability during and after a nuclear holocaust. Additionally the amount of growth in the internet is progressing rapidly. In January of 1995, based on data collected over the previous four year period, the routing tables in the core routers of the internet were increasing by about 17 routes per day. These routes represent networks and not individual host IP addresses. The host addresses were increasing at about 3000 per day [9]. Between mid 1994 and late 1995, the number of domains and address prefixes (routes) had nearly doubled in size.

No one associated with the ARPANet project in the early 1970's would have envisioned the extreme growth and acceptance the internet has gained. Growth factors have forced the introduction of the Autonomous System (AS) concept because the internet address space could no longer be managed as a flat system. Additionally, the internet Protocol version 6 (IPv6) is being designed to increase this address space because IPv4 is running out [24:97]. In IPv4 the address field is 32 bits long allowing for, in principle, 2^{32} (> 4 billion) different addressable entities. In IPv6 the address field is increased to 128 bits, or enough for 2^{128} addresses. Internet design by redesign, large scale internet use, and increased reliance on it to carry daily business operational data leaves vulnerabilities, the exploitation of which can have increasingly devastating effects.

Air Force internet (AFIN) traffic is not immune to these vulnerabilities. Once AFIN traffic leaves a base demarcation point, it becomes part of the public internet infrastructure. Approximately 95% of AFIN traffic traverses the public internet infrastructure at some point along its route [15:5]. This makes DoS attacks potentially damaging from any point in the internet. Additionally, many standard base level systems like supply, finance, or personnel, have single large databases where master files are updated. This represents single points of failure for these systems. All an attacker seeking to deny service need do (among many other choices) is to drop an Internet Control Message Protocol (ICMP) bomb such as "destination unreachable", "source quench", or "redirect" on a host, or flood the router serving the host to deny/degrade service for the entire system [23:336-337].

Lessons Learned From the PSTN

A basic goal of a packet switched or circuit switched infrastructure is fault tolerance. This is an essential element in resiliency to DoS attacks. In the internet, DoS attacks are one of the easiest to launch and one of the toughest to defend against [17]. Establishing resiliency against DoS attacks in the internet requires routing systems that, like the switches of the PSTN, are loosely coupled. Loose coupling achieves robustness at the expense of increased complexity in components, allowing a wider range of operating parameters and interactions, i.e. fault tolerance. In the PSTN for example, about half the software in their switching systems is dedicated to error detection and correction [12:35].

The protection mechanisms at the internet switch level are passwords, access lists, and routing tables that provide rules for access and define its operation. Additionally, routers are designed to limit the amount of bandwidth that certain types of traffic like ICMP are allotted to prevent race conditions in the protocol that would soon flood the router with useless messages. Routers though, unlike the PSTN switches use much less error detecting/correcting software and are more vulnerable to DoS conditions.

As an example of the DoS fragility of the routing infrastructure consider that on April 25th 1997 a router glitch at MAI Network Services, an Internet Service Provider (ISP) headquartered in McLean, Virginia, caused widespread congestion and network outages on one of Sprint's main backbones. Outages were felt nationwide and perhaps internationally. "The outage underscored the fragility of the infrastructure that underlies the global network and how easily a problem with one small ISP can be amplified throughout the internet" [25]. The problem manifested itself by routing announcements that caused major portions of eastern U.S. internet traffic to be routed directly through the MAI's routers (or "black holed"). This quickly overloaded their routers and shut them down. At this point the problem should have self corrected in the internet but did not. The main Sprint backbone routers had to be manually reset.

Because of the differences of applications between the PSTN and the internet (real-time voice versus data traffic whose end applications can withstand delay and intermittent interruptions in the data stream) the analogy between the two systems is not perfect. But the differences are shrinking with the advent of newer protocols such as IPv6 and ATM as they are being designed to carry multimedia data that is, like the voice data stream of the PSTN, less tolerant to delay, out-of-order packet delivery, and jitter. Other commonalties between the systems can be observed in the context of DoS. System saturation can make setting up a circuit to handle a call more difficult, and under extreme circumstances, a circuit may not be available. Similarly, data packets at the network level are not guaranteed to make it to the destination leaving it up to the transport layer protocol to handle errors and retransmissions. Given a busy enough network, some applications actually time out which is a situation analogous to the circuit busy of the PSTN. Consider also, that newer switching protocols like ATM are being designed to handle high bandwidth multimedia traffic such as voice and video which cannot tolerate delays are connection oriented (or virtually circuit switched). This is further evidence that lessons learned by an analysis of the vulnerabilities of the PSTN are applicable to the internet.

Since 1992 telephone companies have been required to report outages affecting more than 30,000 customers to the FCC. Statistics from April 1992 to March 1994 show that the most devastating cause of outage in terms of customer minutes lost was system overload [12:33]. The following charts help to focus the severity of the overload condition as compared to the causes of other types of outages. Note that the second most damaging type of failure was acts of nature, which one may expect to actually cause the most damage in terms of customer minutes lost. But the damage caused by acts of nature ran a distant second to overloads.

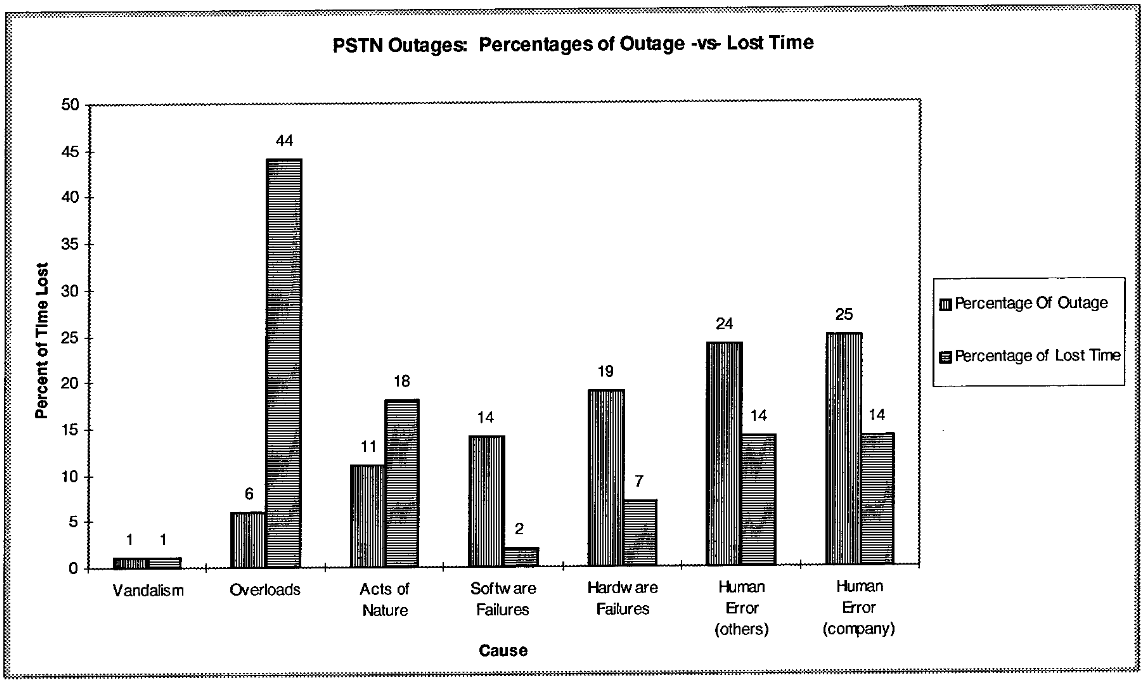


Figure 1. PSTN Outage Type Versus Lost Time

Before explaining the correlations with the internet routing infrastructure, it would help to explain some basic terms. Refer to Figure 2 as needed. As mentioned earlier, the growth of the internet necessitated the introduction of the AS structure to introduce hierarchical routing and get away from the flat routing method where global information was kept at each router. ASs are bounded in that there is administrative control over the structure to handle such things as configuration, security, and route peering relationships specified in the configuration of the inter-AS routing protocol used. A peering relationship is an agreement between two routers to share network topological information that allows the computation and advertisement of least cost routes in the network. Routing protocols within an autonomous system may be different than what is used on the backbone between core routers and between different AS's themselves. This means protocol conversion that creates overhead, delay, and possibilities for error at the AS boundaries. Examples of protocol differences can be such things as quality of service offerings and methods of route computations. ISPs can support point to point connections such as MODEM connections from individuals, or can offer enough bandwidth to support Local Area and Wide Area networks. An ISP is a commercial offering that can support multiple source internet traffic. Individual Local Area Networks however may have the capability to have their own gateway router, administrative staff, and direct leased communications line to an internet Network Service Provider (basically, a larger ISP) therefore being their own ISP. Many individual connections such as those to ISPs or LAN dial-ups go through the telephone company's local loop before getting internet connection whereas LANs and WANs can bypass the local loop. For simplicity, telephone loops and central offices are not shown in Figure 2.

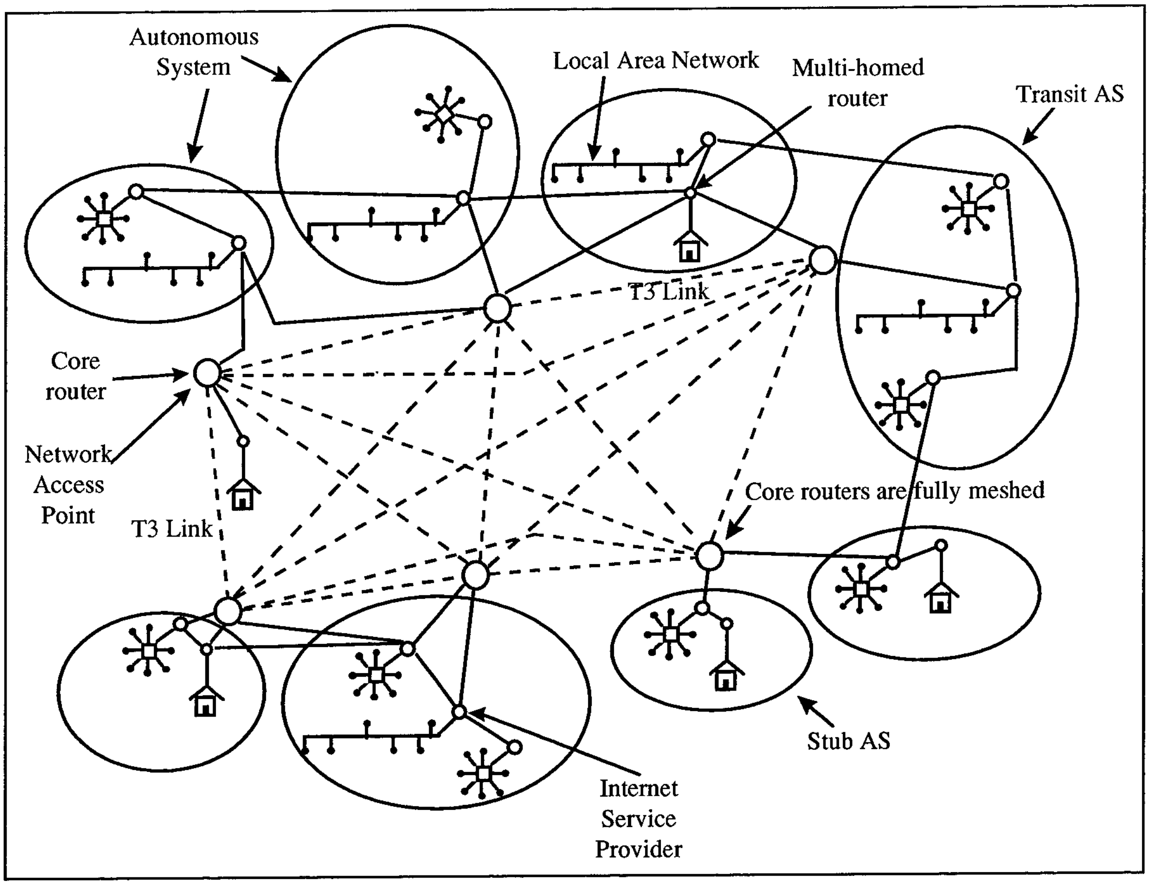


Figure 2. High-level Internet Topological Structure

An Autonomous System (AS) is a set of routers under a single technical administration, using an interior gateway protocol (IGP) and common metrics for best path determination to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs. ASs have a unique number identifying them. This number is used in inter-AS routing protocols to identify the AS from which an update has come. The Border Gateway Protocol (BGP) is the main exterior routing protocol employed in the internet and is known as a *path vector* protocol because it keeps AS information as a route is propagated. Since this classic definition was developed, it has become common for a single AS to use several interior gateway protocols and sometimes several sets of metrics within an AS. The use of the term Autonomous System here stresses the fact that, even when multiple IGPs and metrics are used, the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through and inside it. A transit AS is one that passes packets to other ASs as needed. The transit AS would have an established BGP peering session with other (presumably geographically convenient) transit ASs. A stub AS can only receive packets destined for itself. Also, a stub AS should not normally perform BGP peering with other ASs, but should be statically routed to its upstream ISPs. An exception to this good-sense policy is when a stub AS is multi-homed to more than one ISP. In this case it may choose to accept certain BGP updates from one vice another ISP. To remain a stub AS however, it should not advertise any reachability information but its own.

The BGP is an inter-AS routing protocol. The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of ASs that reachability information traverses. This information is sufficient to construct a graph of AS connectivity from which routing loops may be pruned and some policy decisions at the AS level may be enforced.

The core routers keep global address information in their routing tables. Intermediate routers need only know a subset of this information to route packets to their destination. If the intermediate routers don't have an address reference in their routing tables, the packet is forwarded to a default address of a router higher up in the structure. Subsequent hops are designed to get the packet closer to the target. Normally core routers maintain peering sessions to exchange routing information via BGP with an average of about 32 intermediate sized routers (or routers one tier below core). Of course since all core routers maintain global routing information, they must maintain peering with all other core routers. These core routers are called network access points (NAPs).

The system overload example of the PSTN can be extended to the internet routing infrastructure. In addition to [25], which was an happenstance occurrence, conditions leading to possible overload on a large scale have been found by such studies as the National Science

Foundation-sponsored Routing Arbiter Project recently completed by the Merit Foundation and the University of Southern California Information Sciences Institute. In an analysis of routing stability in which routing data was gathered at the five major US NAPs since January 1996, it was found that there was an inordinately high number of route withdrawals taking place on the core internet routers [13:2]. This caused the routers to "flap", or lose their ability to route efficiently. Flapping is caused by high frequency of routing updates propagating through the routing tables of the routers on the internet. Flapping storms have collapsed portions of the internet.

The number of routing updates per day has grown disproportionally to the number of new routes that are being established. The aggregate level of instability is rising. There are about 45,000 routes on the core routers yet there are between 3 and 6 million routing prefix updates each day [13:4]. Most of these announcements are route withdrawals that are redundant and repeat with a period of about 30 seconds. These are referred to as *pathological updates* in [13]. Preliminary study has produced the following observations:

- The Border Gateway Protocol (BGP), which is the protocol used predominately in the portion of the internet external to an AS and is defined in RFC 1771 [22], specifies an object, "bgpPeerMinRouteAdvertisementInterval" that is defined as "[the] time interval in seconds for the MinRouteAdvertisementInterval timer - the suggested value for this timer is 30 seconds". This object determines the minimum amount of time that must elapse between advertisement of routes to a particular destination from a single BGP speaker. Additionally, the situation worsens in the case of specific route withdrawals. To avoid long-lived black-holes (i.e., packets sent to a host via a non-existent path), the timer does not apply to route withdrawal announcements. The route withdrawal announcements are propagated immediately.

- An analysis of the data shows that all pathological routing incidents were caused by small ISPs. Research is ongoing as to the exact cause(s) [13:6].
- The BGP protocol itself may be acting contrary to its function. BGP peering sessions in the established state are maintained in that state by a keepalive timer. The suggested value for this timer is 30 seconds. If either host does not receive a keepalive message within the timer interval, the connection is terminated. A problem can occur when the internet is experiencing heavy traffic loads. Under heavy loads the keepalive messages may not be delivered within the timer interval [13]. Once the BGP session is dropped, traffic flowing along the pre-existing route has to be re-routed. This means that new routes have to be computed and then advertised down the line. The re-computation of routes is a CPU intensive process that can cause delay in routing live packets and thus queues can begin to fill. This adds to the overhead and can cause widespread congestion thus forcing other sessions to be dropped. Once terminated, all BGP sessions try to re-establish connection periodically until the session is brought back up. Upon establishment of a "new" session, BGP sends full routing information to its peering clients. This means more updates and a ripple effect (route flapping, or a routing storm) is generated throughout the internet. Under normal operating conditions, the BGP protocol sends full information only on start up of the session, then sends incremental updates thereafter, governed by the bgpPeerMinRouteAdvertisementInterval with the exceptions explained in the first bullet above.

The instability is increasing in spite of "fixes" to BGP in the form of flap dampening algorithms and route servers designed to reduce the routing computation load on the core routers. These servers have been deployed at all the Network Access Points. The sole function of the route servers is to do routing table computation for the core routers to free the resources of these routers to do packet switching only. The introduction of route servers was not something that had ever been considered in the design of the internet, it is just another shoring up example or "design by re-design". Table 1 shows a representative snapshot of routing instability caused by pathological routing updates. As seen in the table, redundant route withdrawal announcements can be orders of magnitude higher that the number of IP prefixes contained in the router's table.

Table 1. Partial List of BGP Update Totals per ISP on 1 Feb 1997 at the AADS NAP [13:6]

Network	Announce	Withdraw	Unique Prefixes
Provider A	1127	23276	4344
Provider B	0	36776	8424
Provider C	32	10	12
Provider D	63	171	28
Provider E	1350	1351	8
Provider F	11	86417	12435
Provider G	2	61780	10659
Provider H	21197	77931	14030
Provider I	259	2479023	14112
Provider J	2335	1363	853

The growth and ensuing instability of the internet is creating its own DoS environment regardless of external threats by any individual seeking to initiate a DoS condition. Most DoS attacks seek to cause an overload condition thereby disabling individual links to broad sections of the internet. In the light of DoS conditions being reached under normal operating conditions, it is disquieting to consider the implications of a DoS attack that could cause system wide degradation and outages by propagating invalid routes or otherwise seeking to corrupt switch control mechanisms.

Recent widespread outages caused by seemingly innocuous events have heightened survivability awareness in the community. In an attempt to compensate for the weakness in the infrastructure, newer protocols are being designed with explicit security capabilities. The proposed Simple Network Management Protocol version 2 (SNMPv2) will make use of hashing with shared secrets to guarantee authenticity and integrity of its messages. This is important because it is introducing protections of the control of the switching fabric. This will deny attacks against the SNMP protocol that target the routing environment assuming the shared secret can be protected. But the adoption of SNMPv2 is not global, and protocols such as BGP remain unprotected.

Implications of DoS Attacks

The Air Force core competency of information dominance as it applies to this study can be considered in the light of seven criteria that determine the quality of information as defined in Joint Pub 6-0: Doctrine for Command, Control, Communications, and Computer (C4) Systems Support to Joint Operations.

- Accuracy: Information that conveys the true situation
- Brevity: Information that has only the level of detail required
- Completeness: All necessary information required by the decision maker
- Relevance: Information that applies to the mission, task, or situation at hand
- Security: Information that has been afforded adequate protection where required
- Timeliness: Information that is available in time to make decisions
- Usability: Information that is in common, easily understood format and displays

Of these seven attributes, three can be directly influenced in the electronic channel. Accuracy can be compromised if there are no means to ensure the integrity of the data during transmission. That is, ensure that the data has not been altered en route. Security can be compromised if either the information has been altered during transit or has merely been intercepted and read. Assuming encryption of data can provide security ergo accuracy, timeliness can still be affected by overall system degradation or traffic interception and delay, both forms of DoS attacks. The influence of these three attributes can also cause secondary damage to the quality of information by degrading the other four attributes.

Examples of DoS Attacks

To motivate the measures designed to counter DoS attacks, it is instructive to look at how they are constructed. They can be directed against a single user, system, or can target the infrastructure. Each has its own level of duration, distribution, and disruption. Although DoS conditions can be achieved through an attack aimed at a specific application, operating system hole, or user, this research focuses mainly on the network layer of the OSI 7-Layer model. As such, it is concerned more with analyzing the vulnerabilities of protocols and system holes at that level. Presented here are some representative examples of such attacks. These illustrations are not considered to be all inclusive. But they are indicative of the types of attacks mounted against an infrastructure and will serve to suggest suitable countermeasures (presented in the next section).

In an attempt to characterize the relative damage done by various DoS attacks, it would help to have an common terminology. The study of the vulnerabilities of the PSTN suggested a framework that could be used to standardize the threat in terms of potential damage. According to the study, information systems attacks can be described in terms of duration, distribution, and disruption [6:31]. The following discussion of various forms of DoS attacks will use these ideas.

- ICMP Bombing: ICMP can be used to re-route traffic on the fly. Routers use this to tell hosts that a destination host is unavailable. An attacker can send an ICMP "*destination unreachable*" command to a host to knock the destination off the air as far as that host is concerned. The duration and disruption factors are achieved but the distribution of DoS is limited to the particular host mentioned in the attack. The lack of protection against ICMP bombing has led many administrators to disallow ICMP messages through their firewall. Of course, this may mitigate the good uses for which the protocol is designed.

- Syn/Ack Attack with IP spoofing: This attack can be used to knock a host off the air temporarily. This type of attack is effective against World Wide Web (WWW) servers. The session set up is achieved through a three way handshake between requester and host. The requester sends a *syn* message to a host. The host returns with a *syn ack* to the requester. The requester then sends an *ack* message after which the session begins. A Syn/Ack flood attack is achieved when the attacker sends a request to begin a session to a host and spoofs the originating IP address by replacing it with a bogus one. The host responds to the *syn* message by sending a *syn ack* into the ether. The *ack* never follows. This leaves a *half-open* state at the server. These half-open connection states are saved in a service queue at the server that can become full. Now when legitimate users want to establish a session, the server cannot respond because its service queue is full. Disruption of service is achieved. The duration is questionable because servers flush their half-open connections from the service queue periodically, though an attacker could prolong the disrupted state by sending spoofed *syn* messages faster that the queue is flushed. Distribution is limited to the host attacked. This attack can be thwarted on the sending end by packet filtering. A site can disallow traffic to exit that has a foreign IP address. On the receiving end, a site can only thwart this attack by allowing traffic from trusted sites. But this is a form of self-inflicted DoS, especially if the target is a Web server that is whose information is intended for a large audience.
- ICMP Echo: An attacker can start a race condition between two routers that effectively removes them from service or severely hampers their routing capabilities. ICMP *echo* and *echo reply* messages are sent between routers to request availability status. This is good for maintaining connectivity to and supporting the soundness of dynamic routing. That is, if a router has gone offline, then other routers have to reroute current traffic around the failed device. An attacker initiates the attack by sending router B an *echo*

request where the originating address is not the attacker's but the address of router A. Router B responds with an *echo reply* message to router A. Router A, not having sent the message interprets it as an *echo* request and sends an *echo reply* to router B whereby router B interprets it as an *echo* message and sends another *echo reply.* Thus a circular race condition is established that can quickly eat up available bandwidth and processing capability. The similarity of the messages in the ICMP protocol allows the misinterpretation. This is a more serious attack because in addition to causing disruption with duration, the distribution is achieved against all traffic passing through the routers. A countermeasure is to not allow ICMP traffic to chew up more than X% of the bandwidth or processor load. This is reasonable and will not interfere with legitimate requests because outside of a race condition, the amount of bandwidth used by the protocol should never climb above the noise level during normal operation.

- Traffic Replay, Delay, and Bogus Traffic: An attacker can attempt to flood a router by capturing streams of data and then replaying them later. This causes extra burden on the routers by having to make routing decisions on defunct traffic. If the link is flooded severely enough, the router's ability to switch legitimate traffic could be hampered. The same idea works with fabricated traffic as well. With the former, it is up to the transport layer protocols to reject the traffic. With the latter, the traffic may stay in the system until the time to live field (referred to as the Hop Count field) of the IP header is decremented to zero. The time to live field of a packet is decremented by one for each time it passes through a router. If no home is found for the packet before its number of hops are exhausted, the packet is discarded at the router where the hop count was exceeded. Delaying traffic may cause overhead at the network layer by causing delayed acks and retransmissions because transport layer protocol timeout values have been reached. This adds unnecessary traffic to the links. The disruption, duration, and distribution effects of these types of attacks are hard to gauge and may serve just to add noise to the internet because these types of attacks are aimed at individual sessions.

- Attacks Against The Control Structures: This attack is more sophisticated as it requires knowledge of the operating system of the routers, their internal control mechanisms and structures, and passwords. However, armed with such knowledge an attacker can use SNMP commands to remove a router from service, or can cause bogus BGP information to be written into the routing tables that will promulgate bad routes and could have the effect explained in [25] thereby jeopardizing entire sections of the internet backbone. Since no provision yet exists to guarantee authenticity, integrity, and confidentiality, all an attacker needs to do is use a sniffer set to filter SNMP traffic and thus capture the full session including passwords. In the case of BGP traffic, one could obtain access to a trivial FTP server (where copies of router configurations are kept for ease of loading) and corrupt the information causing the next load of the router from that server to be bogus. Because of the current strain on the internet backbone with frequent brown out conditions occurring under heavy traffic loads, this type of attack is potentially devastating. Disruption, duration, and distribution are maximized. As stated above, current protocol development includes measures to ensure the authenticity and integrity of SNMP traffic with the introduction of SNMPv2 that proposes the use of hashing with shared secrets and encryption. In the case of the BGP protocol, the specification has made allowance for a security association between two BGP peers using a shared secret with hashing to ensure authenticity and integrity of the BGP message at the transport level, but this it not used in practice [14].

Insuring Survivability in the Presence of DoS Attacks

The countermeasures designed to build a robust network layer are those that provide for the requirements of authenticity, integrity, confidentiality, non-repudiation, and access control [10]. Below each is explained along with the mechanisms used to achieve them and the rational behind the requirement.

- Confidentiality: No one but the intended recipient shall know the contents of the message. This is achieved through encryption. Common encryption mechanisms are digital signatures and the Data Encryption Standard used by the federal government. This countermeasure is employed when it is necessary to conceal the contents of the message which may mean just the data or it may also include the encapsulations of the various protocols at the different layers of the OSI model. Employing encryption at the network level for all traffic is not feasible because of the overhead of decrypting the traffic at each router. Although the Secret IP Routed Network (SIPRNet) engineered by the Defense Information Systems Agency does encrypt the full datagram and can operate at T1 speeds, it is expensive in terms of hardware as encrypting/decrypting devices have to be deployed at each routing point. So outside of this network that is designed to transport top secret information for the DoD, encrypting at the packet level for the "standard" internet is not feasible today. IPv6 has a provision for an encapsulated security header, but this is designed to be an end to end protocol conversion. In this protocol, once the packet is encrypted, it is re-wrapped with the routing information so that routers won't have to decrypt the destination address and other pertinent data at each hop. Presently confidentiality countermeasures work well with applications that have built in delay at the application level and are not real time. This will allow for the extra overhead of decryption. An example is e-mail. However to prevent sniffing of SNMP traffic (which carry passwords that allow access to the router), confidentiality countermeasures need to be employed unless a decision is made to accept the risk. Note that knowing the contents of SNMP traffic to include access-control passwords won't do an attacker much good in the case where authenticity and integrity countermeasures are deployed.

- Authenticity: The receiver shall be able to reliably identify the sender. This is accomplished through hashing the contents of the packet along with a shared secret between the communicating parties. The industry is now leaning toward using the MD5 hashing algorithm. In it a unique 128 bit message digest is computed from successive 512 bit blocks of message where the result of the last round is used as input to the next round. The first thing that is hashed is a shared secret. The digest is then appended to the packet. At the receiving end, the hash is recomputed and the two digests are compared. If they match then the packet is assumed to be authentic. Presently, MD5 speeds are compatible with LAN speeds, but is still too slow for general application level traffic on the internet without the use of specialized hardware.
- Integrity: the recipient of a message shall be able to verify that a message has not been altered en route. The discussion for authenticity also applies here. But other measures are needed to protect against replay and delay of legitimate traffic. To prevent replay, sequence numbers have to be used. This means that this information would have to be included in the hash. Delay prevention is much more complicated. To compensate for delay, a time value has to be hashed. This means that some type of global clocking scheme will have to be used with acceptable deltas for clock drift between systems.
- Access Control: Support for controlling communications among elements of a routing infrastructure. The authenticity and integrity countermeasures are a stronger countermeasure. Access control is used today between routers to set up BGP peering sessions, and once set up, to determine the scope of BGP information that will be passed.

Access control uses the IP address and AS number as an identifier so this countermeasure is susceptible to spoofing.

- Non-Repudiation: Prevents a participant in a communication from later denying participation in that communication. That is, there is strict accountability. The authenticity countermeasure described above is sufficient to support this requirement.

A stronger case can be made to protect the control traffic on the internet, the traffic that controls the function of the routers and the promulgation of valid routes. The benefits of protecting the application traffic are in question because of the extreme overhead associated with hashing each packet and the questionable effects that a DoS attack can have when the target is just one session or perhaps one host. The point of this research is to model scenarios in which the countermeasures of authenticity and integrity are employed for protocol traffic. Modeling will help to characterize the operation of the internet when these countermeasures are deployed and will show the feasibility of such a deployment.

Related Research

Research applicable to the current routing instability is being conducted by Merit Network Systems, Inc. under the auspices of the National Science Foundation Grant NCR-9321060. This research is concerned with internet performance measurement and analysis. It will be used as background for this research because it provides information on the daily operational characteristics of the internet. The information can be used as a baseline from which an assessment can be made about the capability of the fabric to accept the extra overhead of introducing explicit survivability components. As shown by the PSTN example, survivability is primarily affected by the correct operation and robustness of the switch control within the infrastructure. In the light of the Merit study, it will be necessary to characterize extra burden introduced to the infrastructure by having routers verify authenticity and integrity of the internet control traffic.

Current work specifically addressing the survivability of the infrastructure is being conducted under contracts from the Defense Advanced Research Projects Agency (DARPA). BBN Systems and Technologies is performing survivability requirements analysis and countermeasures design and analysis with the goal of internet infrastructure protection. The BBN work has identified the requirements specified above (authenticity, integrity, confidentiality, access control, and non-repudiation) as being sufficient to provide for infrastructure survivability. The decision to support these mechanisms in the current IP environment balances risk factors against the overhead of extra processing required to meet the requirements. BBN has specified the requirements but has made no recommendations as to specific deployment mechanisms based on an analysis of the infrastructure to tolerate the overhead. The purpose of my research is to model the introduction of the authenticity and integrity countermeasures for protocol traffic at the network layer of the internet.

Summary

Information dominance occurs when U.S. decision makers possess the information required to make decisions faster and better than any enemy [4:14]. This idea as presented in the AF 2025 study led to the introduction of the Information Superiority core competency put forth in Global Engagement: A Vision for the 21st Century Air Force. DoS attacks directly assault this core competency. Tools to affect the survivability of the network can mitigate the assault on the U.S. edge in information superiority. In chapter three, the chosen methodology to approach the survivability research issue, which is a modeling and simulation approach, is explained. Also, rationale is given that defends and attacks the methodology. This is done to temper the data gained and conclusions derived from the research so that the reader can be given an overall frame of reference from which to view the results. Chapter four explains the model chosen to represent the problem domain. It also gives a full explanation of the properties of the model to motivate its validity in representing the internet survivability research platform.

III. Methodology

Introduction

Before a methodology can be defined that could be shown to have reasonable measures of verification and validity, the approach to the survivability of the internet infrastructure has to be put in the appropriate context. The current infrastructure undergoes route flap storms by the re-announcement of previously withdrawn routes and does so on a large enough scale and with a small enough periodicity to cause high levels of router CPU utilization, cache misses, and routing table recomputation. As was shown by data in chapter two, the current operational characteristics of the internet are growing more unstable as the internet grows. This has manifested itself in increased diameter and node degree, a more loosely stratified hierarchy, traffic delay, delay in route convergence, and anomalous protocol behavior causing brown/black out conditions on sections of the internet. Furthermore, since April 1995 this growth is occurring outside of the heretofore somewhat controlled growth afforded by the administration of the internet backbone by one entity, the National Science Foundation. In addition, the commercialization of the internet and its relatively new popularity with WWW services that include data, audio, and video is creating an environment where there is fast growth in the fielding of new ISPs. This growth is occurring via market demand rates but not necessarily in an efficient manner. Also, the pattern of IP traffic has changed because of the evolution of internet services. Today's traffic is more bandwidth intensive and more inter-AS based than it has been. Changing traffic patterns, explosive internet growth, and a heterogeneous mix of ISPs who realize different routing policies and implement their services with varying degrees of expertise are the culprits in the internet's declining infrastructural integrity. So, ironically, the measures to inject resiliency against DoS attacks against the infrastructure have to be considered within the framework of a current infrastructure that is experiencing "natural" DoS conditions in the forms of brown outs and flapping storms. The deployment of the hashing countermeasure against the DoS threat has a greater potential for having the opposite effect than desired when injected into an ailing infrastructure.

Motivation of a Modeling/Simulation Methodology

Two measures of the infrastructure that can correlate growth of the internet with a decrease in the survivability of the fabric are route stability and topology. By collecting two chronological traces of routing transitions that consisted of 11.7 million BGP updates heard at two core routers [7:5], one study has been able to characterize these attributes:

- Route stability affects reachability to address prefixes. The study measured *prefix availability* (the fraction of time that a prefix is reachable), and *prefix steadiness* (the mean duration of all intervals in a snapshot over which the prefix was continuously reachable). The data for this was gathered from two separate 21-day snapshots, one in November 1994 and the other in May 1995. In the first snapshot, 90% of the prefixes are available for more than 99% of the time, in the second, the availability figure drops below 97%. A similar trend is observed in prefix steadiness. In the first snapshot nearly 99% of hosts are available for more than 99% of the time, but in the second, only 95% of the hosts are available for 99% of the time.
- Topologically, the internet has remained fairly constant in terms of degree and diameter. The data for this was gathered from three snapshots. The first two are identical to the two above and the third was collected from a 21-day period in November of 1995. The average node degrees changed from the three snapshots (in chronological order): 2.67, 2.68, and 2.99. The degrees were obtained from the number of BGP peering sessions per domain. Also, the diameter has remained nearly constant. The diameter, which is a

measure of number of domain level hops of the maximum hop-count between any two domains, was 9, 10, and 10 for the three snapshots, again in chronological order.

For the purposes of motivating a methodology to characterize the current survivability of the infrastructure however, two important topological aspects that could be used in a model of the internet from which a simulation could be run, are degree class and hierarchical connectivity by class. The data suggests a degree distribution that roughly makes up four levels of domain classification summarized by the following table:

Table 2. Internet Degree Classification [7:10]

Class	**Degree Range**	**Approximate Fraction of Domains in Class**	**Types of Domains**
C_1	≥ 28	0.9%	National Backbones
C_2	10-27	3.1%	Large Regional Providers
C_3	4-9	9%	Smaller regional providers, and large metropolitan area providers
C_4	1-3	87%	Smaller metropolitan area providers and corporate or academic networks

As for the connection hierarchy, the internet has a significant portion of links that do not connect a class of router directly above or below the present class. There is a decided non-hierarchical connectivity between classes as shown in the following table. An element of the matrix indicates the fraction of the total number of links that exist between the corresponding classes. The data for this was obtained from the third snapshot.

Table 3. Connectivity Between Classes [7:13]

	C_1	C_2	C_3	C_4
C_1	0.012	0.053	0.064	0.250
C_2		0.030	0.059	0.236
C_3			0.034	0.164
C_4				0.098

The data also amplify the findings in [13]. Recall that all pathological routing update behavior was observed as originating from small ISPs. A correlation to the increasing instability of the internet is drawn from the fact that address prefixes have grown at a slower rate between the three snapshots than have domains and links. If the November 1994 snapshot is taken as a baseline where there were 531 domains, 709 links, and 21524 prefixes, then the growth of the internet in the two subsequent snapshots can be given as a percentage of the baseline. In the second snapshot, there were 746 domains, 1000 links, and 26945 prefixes for a growth of 140%, 141%, and 125% respectively. In the third snapshot, the figures increase to 909 domains, 1369 links, and 31470 prefixes for a growth over the baseline of 171%, 193%, and 146% respectively. This suggests that smaller ISPs represent an increasingly larger percentage of all ISPs over time.

The methodologies employed by the current research in characterizing the internet infrastructure have largely been based on empirical observations. The studies in [13] and [7] are based on collection of historical BGP data from the national backbone routers at various network access points. Further, these have been focused on the current ability of the infrastructure to operate correctly under heavy usage patterns with errors introduced by router misconfigurations and less than robust deployments of BGP. BGP implementations have been found that are stateless and whose internal timers are un-jittered. The stateless BGP implementations can more readily lead to the transmission of redundant routing information, and the static timers can lead to self-synchronization between routers over large areas. In a synchronized state these routers could transmit large amounts of BGP data almost simultaneously leading to increased use of bandwidth and very heavy CPU utilization. This will hamper the router's ability to actually route packets. This leads to longer queues, delayed user and protocol control traffic, and with the dropping of BGP sessions, could lead to an escalating state of unsteadiness and route storms (or flapping).

The larger vendors of routing equipment have their own live test beds in which to model the behavior of current and proposed protocols in which they can control the different facets of the protocol that is being modeled and can control the configuration of their test bed. In the absence of a live test bed, this research will focus on the *symptoms* of current internet instability by modeling a representative network structure that conforms closely to the topology of the current internet in node degree, diameter, and connection hierarchy. These symptoms include amount of delay in the system from a link and end-to-end reference, the amount of CPU utilization on router objects, the amount of queue utilization, system loads, and bandwidth utilization. It will also attempt to show an escalating pattern of instability with traffic load.

Data from the live internet points to decreased performance with growth. However it also shows cyclical performance degradations with increasing traffic loads on a daily basis [13:7]. The routing instability is greatest between noon and midnight for times corresponding to North American daily traffic. Three types of routing instability were measured:

- A route is explicitly withdrawn as it becomes unreachable and is later replaced with an alternative route to the same destination. The Alternative route differs in its ASPATH or nexthop attribute information. This is a type of forwarding instability. *The ASPATH is part of a BGP update message. After each router makes a new local decision on the best route to a destination, it will send that route, or path information along with accompanying distance metrics and path attributes, to each of its peers. As this reachability information travels through the internet, each router along the path appends its unique AS number to a list in the BGP message. This list is the route's*

ASPATH. An ASPATH along with an address prefix provide a specific handle for a one-way transit route through the network [13:2].

- A route is implicitly withdrawn and replaced by an alternative route as the original route becomes unreachable, or a preferred alternative path becomes available, This is a type of forwarding instability.
- A route is explicitly withdrawn and then re-announced as reachable. This may reflect transient topological (link or router) failure, or it may represent a pathological oscillation. This is generated by either forwarding instability or pathological behavior.

This data gives an indication that the main factor to be controlled during a simulation of the survivability characteristics in the internet is traffic load. This seems like a natural hypothesis because the study will be run to ascertain the viability of introducing measures to thwart DoS attacks. These measures will bring their own overhead and will introduce their own loading characteristics in terms of CPU utilization and throughput. Since these measures will be deployed for control traffic, it will be necessary to know what percentage of the overall traffic is represented by BGP traffic.

A secondary consideration as a simulation control parameter will be the assignment of priority to BGP messages. The specification for BGP4 [22] suggests that BGP messaging be supported with a version of TCP that is capable of transmitting with priority. This would increase the probability that even during busy traffic periods, BGP sessions would not oscillate because of non-receipt of keepalive messages. Recall that BGP session oscillation can lead to flapping storms because the protocol is designed to send a full routing table update upon session establishment (see the third bullet under the section Lessons Learned From the PSTN in chapter 2).

Modeling Considerations

The growth of the internet has caused irrational protocol behavior and is accompanied by fragmented administration and different protocol implementations by different vendors. But a model that can be built and simulated is usually much smaller and operationally constrained than what is actually being modeled. "The study of algorithms and policies to address [internet problems] often involves simulation or analysis using an *abstraction* or model of the actual network structure and applications. The reason is clear: networks that are large enough to be interesting are also expensive and difficult to control, therefore they are rarely available for experimental purposes" [26:594]. The challenge is to create a model that is representative of the network being simulated. The approach recommended in [26] is needed because prior research using randomly generated models that do not necessarily represent "real" networks have reached conclusions about the suitability and performance of (an) attribute(s) being simulated that is widely variable. The research in [26] introduces a "Transit-Stub" model of internetwork topology that is partly randomly generated but adds components that are designed to mimic the characteristics that are found in real internetworks such as average node degree, diameter, and locality of reference (there is a higher likelihood that any node will be connected to its "closer" neighbors in a graph than those farther away). The Transit-Stub graph generation tool is based on the AS structure of the internet and generates graphs in a hierarchical fashion unlike purely random graph generation tools. The transit ASs act as conduits for BGP routing information they receive by passing that information along to neighboring ASs thus propagating reachability information while the stub ASs do not.

Since this research focuses on wide area networking whose exact topology is not known, the Transit-Stub graph reflects the known properties of the internet and instantiates the remainder of the topology in some random but reasonable fashion. Similar to the attributes measured in [13] and [7], the study considers the following characteristics that are quantitative abstractions of some aspects of the structure of the internet. For a graph with *m* nodes and *n* edges, it considers the following:

- Node degree distribution: the average node degree *2m/n*. As noted above, data from 1995 points to an average degree of approximately 2.99.
- Hop-depth distribution: the hop-depth at node *u* is the depth of the shortest-path tree rooted at *u* to all other nodes. This is another form of internet diameter that is measured at 10 from the 1995 data.

Within a modeling environment then, the topological considerations addressed by this model are representative of the internet structure.

Benefits of a Modeling and Simulation Approach

While data derived from the real internet is authentic, it may not be indicative of the system as a whole and may point to problems more endemic to the hierarchical layers from which the data is gathered unless intelligent filtering is done. Gathering data from the live internet from which a global view can be seen immediately, or reasonably extrapolated, would be prohibitive because of its volume. In a modeling approach however, if the model is somewhat reasonable, a global picture can be easily obtained.

The data gathered in [13] and [7] is assumed to be representative of the global internet because it was gathered largely from core routers. The core routers have global reachability information and process the largest percentage of traffic on the internet. Furthermore, the routing information computed at these routers is an amalgamation of approximately 99% of the topological picture, disregarding connectivity within the class, that because of the vital nature of the core function is assumed to be stable. Note that connectivity between core routers accounts for about 1% of the connectivity in the internet (see Table 3). But the data does not include information pointing to the operating characteristics of the components in the internet subhierarchy. It does not show the number or percentage of lost BGP peering sessions or dropped packets. It also does not capture information about the local operating environment of the "small ISPs" from which the pathological routing behavior originates. Using an abstraction of the real internet in a modeling approach can provide simulation information in finer granularity.

Aside from the obvious benefits of economy of scale, quicker set up and reconfiguration capabilities, inexpense, and availability, a modeling tool can also give various views of the results. Probes can be strategically placed to indicate the operating characteristics of any part of the system. This gives a more robust representation of the systems as a whole or any sub-component. Dependencies can be more fully exploited and a more deterministic line between cause and effect can be shown.

Statistical acquisition is supported (depending on the particular tool being used) by one of three methods. Independent replication of the simulation varying the global seeds, batch mean samples within the same simulation (where the simulation reaches steady state before acquisition starts), and regenerative sampling where successive visits to a system state in interest are sampled. The first two are easy to apply, and where the number of samples is greater than or equal to 10, samples can be assumed to be of a normal distribution with little loss of accuracy. Although the theoretical foundation for batch mean sampling is weaker than either of the other two methods and successive samples cannot be assumed to be wholly independent, in practice its performance has been found to be superior to independent replications or regenerative sampling. Additionally, sampling from regeneration points, although independent, can be difficult in a complex network because those points can be so far apart that it can be impossible to simulate an entire cycle [19,3-35]. This research will use both successive and batch mean sampling.

Shortcomings of a Modeling and Simulation Approach

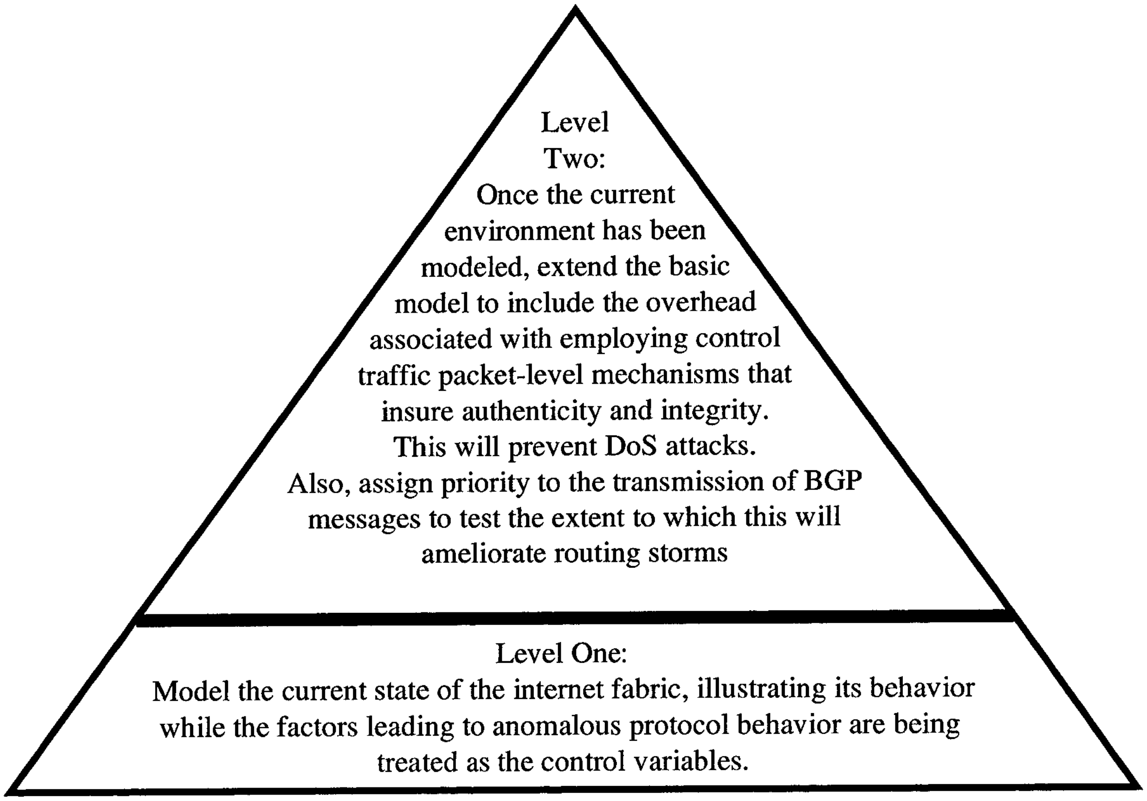
The most glaring drawback to the modeling and simulation approach is attaining a valid representation of the system and parameters under study. The level of abstraction required to implement a reasonable model may skew the results of the simulation. Generic network modeling and simulation tools do not actually model the operation of a protocol such as BGP, they model a representation of them. Primitives are provided to model CPU utilization, and transmission and propagation delays in the form of absolute data structure delays or resource allocation primitives. It is left up to the designer to obtain information about the internal protocol operation so that accurate primitives and parameters can be incorporated into the model. Also, only part of the internet protocol stack is normally expanded, say a particular algorithm, and then studied within the layer where it belongs. Other layers are abstracted to a larger degree. The operation of the internet can only be notionally represented in a model because of its complexity and the size of the system. Even among seasoned network engineers, factors producing less than optimal conditions have to be analyzed in more of a vacuum than is present in their real world deployment. Their complex reticulate structure makes it difficult to accurately reflect their behavior within the system because relationships with other protocols and resources are not understood on a level where an in-depth analysis can be performed.

Adding to the difficulties is an incomplete picture of the data that bears on this research. SNMP data such as dropped packets, amount of BGP traffic compared to application traffic, number of dropped BGP peering sessions (categorized by time of day or contrasted against amount of traffic), are not generally available. If this type of information is available, it is representative of the top hierarchical layer of the internet, the NAPs. Smaller ISPs tend not to release data about their internal networks [11]. Part of the reason may be commercially driven, and part due to ignorance in some cases, and lack of ability to collect and analyze their own data in others. This reflects a main premise of the research: the security paradigm is much more prevalent than the survivability paradigm. In this case, the smaller ISPs are more security than survivability oriented. Indeed, some factors such as multihoming smaller ISPs directly to top level ISPs, while providing redundant service for the smaller ISPs, also present configuration problems and complicates the structure. The probability for configuration error increases and the load on the upstream ISPs increases due to larger numbers of BGP peering sessions. That is, in trying to ensure higher levels of service for themselves, the smaller ISPs are putting strains on the bigger system. This is an example of the "single-system" paradigm again, (which is the paradigm used in designing security measures), where individuals do not equally consider the ramifications of their actions within the framework of the whole system.

Framework and Research Foundation

The model will be used to gauge survivability of the internet fabric mainly from the perspective of the network layer of the OSI 7-layer communications model. It focuses on the most prolific of the protocols that comprise the control fabric of the internet (the BGP protocol), its current operation, and its vulnerability to DoS attacks. It also is concerned with the capability of the communications infrastructure to accept the extra overhead of hashing this protocol to provide authenticity and integrity to individual messages in an attempt to make the protocol immune to DoS attacks. The current protocol that controls the configuration of the internet routing infrastructure is the BGP. BGP is mainly an inter-Autonomous System (AS) protocol that advertises reachability information to all border AS routers on the internet. It is a distributed asynchronous path vector algorithm that determines least cost paths between nodes. It is widely believed to converge to a single set of optimal solutions based on a metric used to represent shortest path. This metric is implementation specific and could represent any number of user-controlled factors such as throughput, capacity, cost, length, or other path characteristics. The computation of the algorithm is local and is relies on the veracity of the medium to promulgate local information in an efficient enough manner so that the locally held information can become part of a globally understood picture. The algorithm is myopic and becomes inconsistent in its view of the world when the internet becomes laden with traffic. In the busiest times, topological information can change more quickly than the protocol can converge. This leads to inconsistent information, temporary routing loops, and routing storms.

One of the goals of this model is to test protocol convergence in an attempt to provide a realistic context in which to model factors mitigating hacker initiated DoS attacks. The research approach has two purposes illustrated by the following figure and then explained in more detail in numbers (1) and (2) below.



Level Two:
Once the current environment has been modeled, extend the basic model to include the overhead associated with employing control traffic packet-level mechanisms that insure authenticity and integrity. This will prevent DoS attacks. Also, assign priority to the transmission of BGP messages to test the extent to which this will ameliorate routing storms

Level One:
Model the current state of the internet fabric, illustrating its behavior while the factors leading to anomalous protocol behavior are being treated as the control variables.

Figure 3. Foundation of the Research Approach

1. The purpose of the first level is to model two postulated reasons for the anomalous protocol behavior (level one in Figure 3).

A. The first is the behavior of the protocol under heavy traffic loads. One-hop Autonomous System neighbors can maintain BGP sessions with each other. These sessions are maintained by keepalive messages that are sent approximately every thirty seconds. If a keepalive message is not received within about 90 seconds, the peering session is terminated. Suitable route advertisements are sent out to remaining neighbors advising of new reachability information concerning the now non-available peer. This causes extra processing on the remaining connected and downstream peers as they update their local routing tables. The protocol now tries periodically to reestablish the peering session with the lost neighbor. Upon successful re-establishment, a full routing table update is sent. This is very computationally expensive for the receiving peers and further exacerbates busy traffic rates. As messages queue, more keepalive messages can be undelivered in the timer window and more sessions can fail. This can go on and produce what is known as a route flapping storm. This storm can spread throughout sections of the internet fabric, only subsiding when traffic periods lessen, typically in the late evening through mid-morning hours U.S. eastern time.

B. The second is observed ill behavior of some Channel Service Units/Data Service Units (CSU/DSU) on data lines. Under increased traffic loads, many units have been observed to exhibit lossy behavior which can cause extra overhead on the internet in the form of lost packets (both protocol and application-level traffic) and their associated requests for retransmission.

2. The purpose of the second level is to model the feasibility of introducing a specific protocol mechanism designed to ensure authenticity and integrity of the control traffic at the packet level (level two in Figure 3). Note that even though the BGP protocol has hooks in place for insuring the authenticity and integrity of its messages, that part of the protocol is not implemented in practice [14]. The BGP security mechanisms would be achieved using MD5 hashing that uses a shared secret as the catalyst for the hashing process. However, this technique is not cheap in terms of computing and comparing hashes at each router as BGP messages are verified. The purpose of providing authenticity and integrity to a protocol such

as BGP is to counteract the threat of hacking the control traffic of the internet which, as opposed to hacking individual (non-control) sessions, has the greatest potential for maximized *disruption* of service, and *distribution* and *duration* of that state. The second thrust in the second level is to model the affects of prioritizing BGP traffic. A main cause in the initiation of routing storms is the inability of BGP to maintain its session connections as the data traffic increases on the internet. This session loss is caused by non-receipt of BGP Keepalive messages that get dropped from transmission queues or delayed in the transmission queues for a longer period than the protocol allows before a timeout event occurs. By assigning priority to the traffic, as is suggested in RFC 1771 [22], the research hopes to show the veracity of the protocol even under pathological loading conditions.

The two control variables of interest in this model are traffic rate and level of loss produced by ill-behaving CSU/DSU devices. The traffic rate can be iterated, within the simulation, from light to heavy. The modeling of the level of loss experienced from CSU/DSUs is controlled with respect to traffic rate.

The study is not concerned with application-level traffic and leaves its protection as something to be dealt with at the transport level of the internet. The proposed Internet Protocol version 6 (IPv6) is one mechanism for insuring end-to-end authenticity, integrity, and confidentiality (as necessary) of that type of traffic. This study is also not directly concerned with solving the flapping issues in the internet today.

Summary

The trade-offs associated with a modeling and simulation approach can leave holes in the research. The overall results are only as valid as the model and parameters used to instantiate it. Beyond that, multiple simulation runs must be performed with different global seeds and the results compared for statistical similarity to ensure that simulation transients are accounted for. The results of the simulations will point to what may be done to ensure certain levels of survivability within the network layer of the internet today and will be tempered by the validity of the model. Chapter four motivates the representative nature of the Transit-Stub graph generation tool. It also explains the parameter instantiation of the Transit-Stub Internet Survivability Model (Transit-Stub ISM). Chapter five will detail the baselining of the Transit-Stub ISM and the results of the simulations. To address possible holes in the research, chapter six will critique the model, suggest areas of improvement in the model and in the research methodology, and finally highlight common-sense measures that network administrators can take to ensure more survivable internetworks.

IV. Model Construction

Introduction

In this chapter the Transit-Stub ISM construction and parameter instantiation is discussed from the perspective of their validity as a representation of the real internet. The operation of the model is also discussed. This chapter represents a top-down view of the model and is intended to give a large-view explanation. Conversely, Appendix A is constructed in a bottom up manner, and contains a detailed explanation of the model including a discussion of every module and parameter. Appendix A is designed to be a companion this chapter.

Basic Model Construction

The Transit-Stub ISM is based on Autonomous System (AS) entities in the internet and focuses mainly on inter-AS traffic. The protocols employed within the AS to maintain intra-AS reachability are most often not the BGP protocol, but other protocols such as Open Shortest Path First, or Routing Information Protocol. The Transit-Stub ISM assumes the correct operation of intra-AS routing protocols.

Transit ASs can act as packet forwarders for traffic not originating in their AS and bound for another AS. The packets are forwarded by border routers, or routers configured to be logically on the periphery of the AS. These routers are the first to receive inter-AS traffic. If the packet is not destined for a host within the AS then the packet is forwarded to the neighbor AS who is computed as being the next-hop in the path to the packet destination. If the packet is destined for that AS, the border router knows where to forward the packet based on the specific intra-AS protocol used to maintain intra-AS reachability information. In some cases where a transit or stub AS is multi-homed to one or more ISPs via separate border routers, then traffic can be segregated before being transmitted from the provider. That is, internal reachability information is injected into BGP that is advertised to the service providers. If a destination network can be reached more economically through one border router vice another, then local policy will allow for the proper advertisement of that internal AS information to the outside world.

A stub AS can only receive inter-AS traffic that is destined for a host within that AS. Regardless of the type of AS being modeled in the Transit-Stub ISM, both AS types generate traffic whose destination can be to any host in the model. The traffic distribution will be both intra- and inter-AS based.

The model topology is based on the Transit-Stub Graph Generation Program developed by Dr. Ellen Zegura and others from the Georgia Institute of Technology [26]. It is based on the transit-stub AS nature of the internet. The program, in addition to producing a transit-stub representation, also generates path metrics that are designed to support locality of reference. As implemented in the Transit-Stub ISM, links are assigned favorable metrics based on their capacity. In this case, the link may be physically longer than another, but would still be preferred if it were substantially bigger.

Representative Nature of the Transit-Stub Internet Survivability Model

This and the next section present a detailed explanation of the Transit-Stub ISM. Because of the length of these sections, the following table can be used as a thumb nail reference. It is meant to give an indication of the validity of the model. Where applicable, model traits and parameters are given values and references.

Table 4. How the Transit-Stub ISM Represents the Internet

Trait or Parameter	Value	Rationale/Explanation	Location(s) in Document	Reference
Locality of Connection (trait)	N/A	The nodes of a model are connected to nodes in the same AS, or to ISP nodes higher in the hierarchy.	4th bullet in this section	[26:595]

Trait or Parameter	Value	Rationale/Explanation	Location(s) in Document	Reference
		This allows more accurate protocol modeling.		
Node Degree (trait)	3.2	This value has been observed to be close to the node degree in the live internet. It more accurately portrays the hierarchical structure in the internet. Allows a finer-grained representation of the protocol being modeled.	2nd bullet in this section; Chapter: 3, Section: "Motivation of a Mod/Sim Methodology"; Chapter: 3, Section: "Modeling Considerations"; Table 2; Equation 8	[7:9]; [26:595]
Max Diameter (trait)	11	This value is also representative of the value observed in the live internet. It affects end-to-end delay.	3rd bullet in this section; Chapter: 3, Section: "Motivation of a Mod/Sim Methodology"; Chapter: 3, Section: "Modeling Considerations"; Chapter: 4, Section: "Instantiation of Particular Model Parameters", Sub-section: "Max Hop Count"	[7:10]; [26:595]
Model Hierarchy (trait)	3 levels	Allows packets in the simulation to undergo different processing based on the level that is being traversed. Adds to the validity of the simulation and affects end-to-end delay.	1st bullet in this section; Chapter: 3, Section: "Motivation of a Mod/Sim Methodology"; Table 3	[13]; [7:10-12]; [26:595]
Timeout Value (parameter)	Eq (1)	Allows retransmission of packets at a data link layer level. Approximates TCP overhead.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	
Time to Reconfigure Network (parameter)	40 sec	Represents a baseline of the amount of work that a node in the model is required to do when receiving BGP updates.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[20]
CSU/DSU Load 1/…/5 and CSU/DSU Failure Length 1/…/5	Eq (5)	Mimics faulty Channel/Data Service Units. As loads increase at the link level, these units have been observed to	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[13:6-7]; [1]

Trait or Parameter	Value	Rationale/Explanation	Location(s) in Document	Reference
(parameter)		cause increased bit error rate		
Max Hop Count (parameter)	15	As packets make their way to destination in the model, the hop count is incremented by one for every node that the packets go through. The packet is retired if this count exceeds the maximum. By querying this behavior after a simulation, the protocol behavior can be characterized.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[7:10]; [26:595]
BGPI Processing Delay (parameter)	Eq (6)	This value is set as a proportion of the time it takes a node to recompute the full network (based on its understanding of what the full network is). It is also contingent on the current BGP policy implementation at the node.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[20]; [8]
Mean Link Down to Up Delay (parameter)	Var	This value represents the time it takes a BGP peer to re-establish a failed BGP peering session.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[20]
BGPIK Traffic Proportion (parameter)	9:100	The amount of BGP interim update (BGPI) messages compared to BGP keepalive (BGPK) messages.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[21]
Total Network BGP Traffic (parameter)	< 1% of all traffic	Will be referenced to the total network traffic found as part of the nominal simulation run.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"; Chapter: 4, Section: "Baselining the Model …"	[14]
Node Out Degree (parameter)	Var	Used to help determine the amount of delay packets encounter at BGP nodes while those nodes are processing BGPI or BGPF messages.	Chapter: 4, Section: "Instantiation of Particular Model Parameters"	[20]

The Transit-Stub ISM (see Figure 4) has 50 nodes that comprise 4 major ISP nodes, 2 transit ASs, and 5 stub ASs. Each AS is represented by a particular BGP deployment configuration that supports its policies. The 4 ISP nodes are logical entities and represent a stopping point for the model. In reality, these nodes would be connected to larger upstream service providers and would each belong to a different AS of their own. The model is also representative of the internet in hierarchy, average node degree, and diameter [7:9-11].

- Three levels of hierarchy are being modeled. The top level is ISPs. The second level is larger transit domains ("domain" can be used interchangeably with autonomous system), the third layer are smaller stub domains. Each layer is instantiated with a representative BGP traffic picture and peering policy. The transit ASs need to have BGP peering with other transit ASs to maintain reachability information. The stub ASs however may or may not peer with other BGP neighbors. However to remain a stub AS, it is essential that if it does have a peering session with another BGP neighbor (normally its ISP), that it not advertise any routes but its own to that neighbor. If a stub AS advertises reachability information for any host not in its AS, then that AS becomes transit. In most cases, it is sufficient for a stub AS to be default routed to its ISP. An exception to this is if a Stub AS is multi-homed (usually for insuring robust service in case one path to an upstream provider is lost). In this case, the stub AS may accept reachability information from all connected upstream providers. The amount of information accepted is based on specific address prefixes and is locally configurable. Within this model, the representation of the local BGP policy is achieved through controlling the amount of BGP traffic that is exchanged during a peering session.
- The average node degree of the model is 3.2. It has 50 nodes and 78 edges representing bi-directional links.

- The maximum hop based diameter of the model is 11 and the average hop based diameter is 4.3024. These parameters are useful in representing a stratification factor within the model that can effect the behavior of the BGP protocol during busy traffic times. That is because the operation of BGP is distributed and asynchronous and the phenomenon of flapping takes advantage of this configuration. While significant sections of the internet can be convergent under BGP operation during nominal traffic periods, when the load increases the protocol takes longer to converge thereby effectively reducing the size of any portion of the internet that is convergent. This creates anomalies in the similarity of route information held by routers and forces the BGP update messages to increase in an effort to create a stabilized environment. The decreasing interarrival times between BGP update messages whose substance represents an increasingly dissimilar topology (a topology which is more rapidly varying), creates stress on the routers and reduces the efficiency with which packets are routed. Couple this with the smaller areas which exhibit convergence and the environment is amenable to flapping storms as these increasingly smaller convergent areas try to reconcile their particular view of the current internet topology with the rest of the internet. The stratification and diameter of the model are similar to the real internet therefore more apt to exhibit the same characteristics of the route flapping environment that the real internet does during flapping storms. Since the substance of the BGP protocol is abstracted in the Transit-Stub ISM, evidence of "flapping" will come from a favorable comparison of the flapping environment. The characteristics which point to the formation of this environment are an increase in the number of BGP peering sessions that are dropped, an increase in end-to-end delay, an increase in the number of dropped packets, increases in queue fullness states, and increases in the amount of network topological computations that are taking place at the BGP nodes in the model.

- Locality of connection is realized by the transit-stub network topological representation and the link cost assignment produced by the Transit-Stub Graph Generation Tool. The confinement of node connection to other nodes in the same AS or to ISP nodes allows more realistic modeling of the BGP protocol. The locality of connection doesn't necessarily have to be based on link distance. There could be large distances between nodes in an AS, but their the AS could still be viewed as an self-contained entity by other AS's. Link cost assignments are made in such a way as to allow the most efficient routing of packets between AS's. The topology in which the protocol is being modeled will support level one of the foundation of the research approach, (see Figure 3), more accurately than will a purely random network representation [26:594].

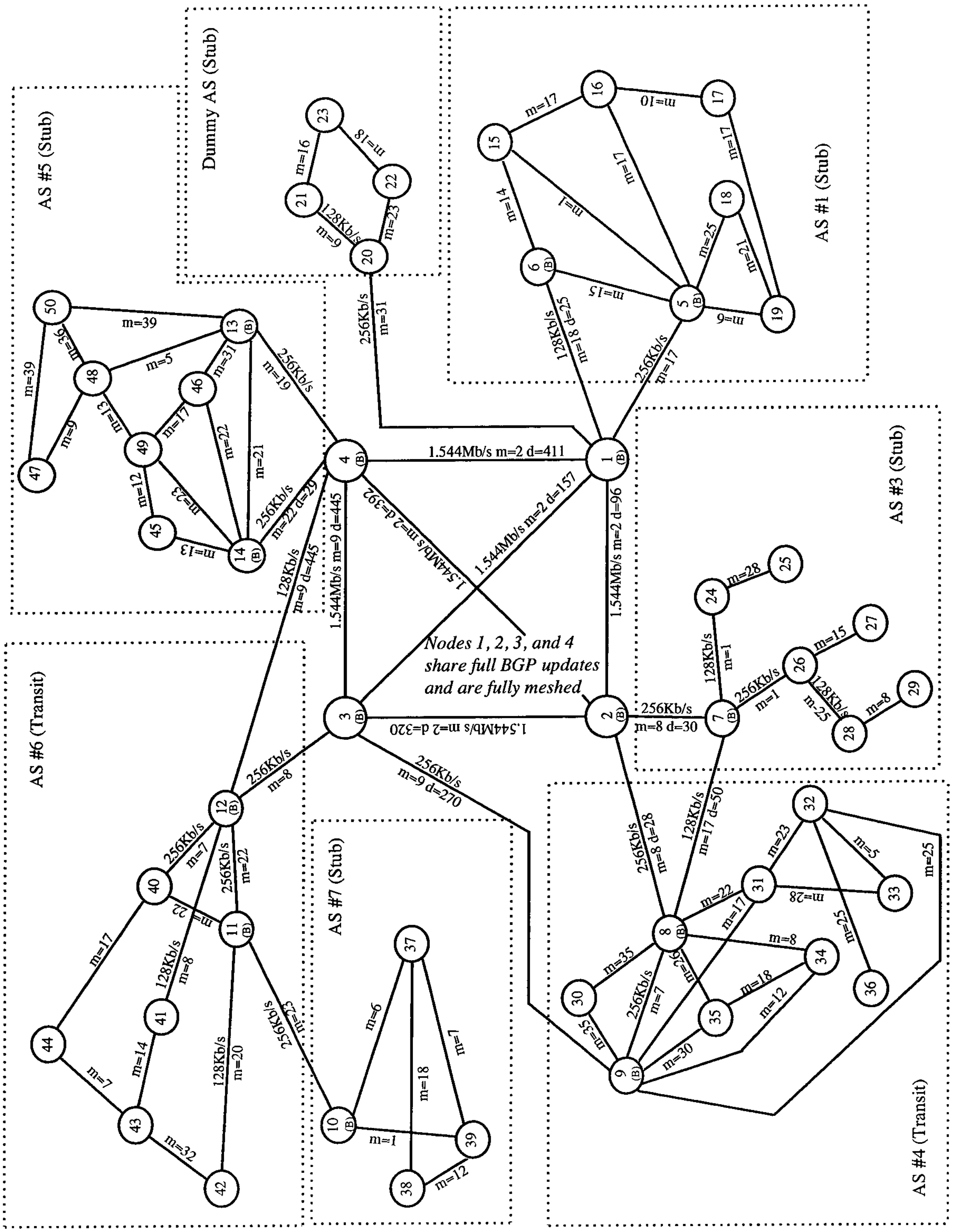


Figure 4. Transit-Stub Internet Survivability Model

In Figure 4, *m* is the link metric and is also the distance of the link in miles unless there is a *d* entry in which case, that entry is the distance in miles. All Link capacities are assumed to be

56Kb/s unless otherwise noted. The nodes that represent border routers and have a BGP processing responsibility within the model have a "B" under their node number. All the ASs are number with the exception of the Dummy AS. The Dummy AS is so small that its full host address information can be kept in the routing tables of its upstream ISP, in this case, node 1. The model is constructed using BONeS Designer from the Alta Group of Cadence Design Systems, Inc. It is based on an example model (the Example WAN Model) provided by them as part of their Modeling Library Reference. I have adapted the model to reflect a representation of the current topology of the internet using the Transit-Stub graph generation tool. That adaptation has also introduced mechanisms to model BGP behavior and CSU/DSU behavior as described in Figure 3.

As was mentioned in chapter 3 in the "Shortcomings of a Modeling and Simulation Approach" section, the actual BGP protocol will not be simulated. That is, the data portion of a BGP update message contains no real information. However, the overall environment can still be modeled because each BGP message has an associated processing delay within the model. These delays are representative of the behavior of the protocol within the live internet today and can be tuned in the model on a node by node basis. And as will be shown in the following section, other model parameters can be instantiated to mimic real internet operation as well.

Instantiation of Particular Model Parameters

Beyond the parameters that are either self explanatory or are artifacts of the simulation itself (see Appendix A for a full explanation of every parameter, resource, event, and memory argument), there are a group of parameters whose instantiation is meant to be indicative of the live internet. An explanation of each of those parameters follow. In cases where live data was unavailable, an argument is given explaining why I chose certain values for them. Also in some cases, justification is given as to the placement of particular parameters within a particular level in the model.

- Timeout value for ACKS: this parameter is computed as

$$TV = TD + 2PD \tag{1}$$

where TV is timeout value, TD is transmission delay, and PD is propagation delay.

Transmission delay is

$$PL/LC \tag{2}$$

where

$PL = Packet\ Length$ (bits)

$LC = Link\ Capacity$ (bits/second)

Propagation delay is

$$LL/C \tag{3}$$

where

$LL = Link\ Length$ (meters)

$C = Speed\ of\ Light$ ($3.0 * 10^8$ meters/second)

Because more there are other factors besides transmission and propagation delays which affect the speed with which an acknowledgment packet is returned (such as CPU utilization or queue states at routers, hosts, or other objects) a positive fudge factor is built into the transmission delay where PL is represented as "max PL + (10 * ACK PL)". In the worst case (an individual packet is the max size), the grace period is still (9 * ACK $PL)/LC$. My model assumes all fiber links in the computation of PD hence C is used, whereas with copper transmission media, $0.67C$ is normally used. This model starts the ACK timer instantiated with the TV after the data packet has undergone its outgoing TD at the data link layer. That is why TV in not given as $2(TD + PD)$.

Normally ACK timeout values (as well as retransmission window sizes) are adjusted dynamically by the operation of the TCP protocol. This is done to decrease the amount of overhead traffic introduced into the links as the network gets more loaded. However this model uses a static representation with little loss of validity. This is because with dynamic ACK timeout adjustment and TCP window size adjustment, the amount of overall traffic change is small compared with overall traffic injection network wide and from a survivability standpoint, is negligible. Also, the precise modeling of ACK timeout values belongs more to the transport layer than the network layer and is session specific.

Since this model will take into account a varying traffic rate to test for saturation, the dynamic adjustment of TCP timeout values and window sizes could be taken as a constant amount representing a decrement to the overall traffic. Therefore iterating through the range of possible traffic loads to test the survivability of the fabric from the network layer perspective is still valid considering that a continuum of traffic load is represented.

As an example consider a link that has an average 55% load during busy hours. At this point TCP could be operating with minimum window sizes and maximum ACK timeout values which would mean that TCP would be adding the very least amount of protocol traffic to the link and the overall network. Say that the corresponding TCP protocol reduction factor is 2%. Subtracting this 2% from the 55% average load has already been modeled however as simulations were run with an iterated traffic load. So from the context of this model, using static values for timeouts and having a sparse TCP representation is still valid from a survivability point of view.

- Time To Reconfigure Network: this value is set at the nodal level and is dependent on the amount of BGP traffic that is processed at the particular node and the router's processing power. In my model, nodes 1, 2, 3, and 4 (see Figure 4) are the highest nodes in the hierarchy and would have both the highest amount of BGP traffic, and the most powerful routers to handle the loads. As an example, a representative router that would be employed at this level would be a Cisco 7500 router that could reconfigure its view of a network within a minute [20].

Within this model, it is assumed that equipment is roughly sized to the amount of work that is required of it. This is a valid representation because, in the actual internet, if equipment were not sized to load, then individual ISPs would not stay in business. But this view is also tempered with the fact that the upper hierarchical layers of the internet today receive attention from various agencies, including government, academic, and commercial, to ensure the correct operation at that level because it is the most vital in the hierarchy. This is fueled by government sector money available from the National Science Foundation which has an historic interest in facilitating the smooth transition of the internet backbone from NSF control to commercial control. Therefore, the representation of the robustness of operation at the upper hierarchical levels will be stronger than at the lower levels within this model. Correspondingly, the parameters controlling the operation of the BGP protocol within this model will be more optimized at nodes 1, 2, 3, and 4.

- ACK Length: ACK packet lengths are 92 bits. The ACK is broken down into 32 bits each for the host and destination IP address and a remainder of 28 bits to represent a sequence number between 0 and (2^{28} - 1). Note that in the actual internet, ACK packets are tied to their corresponding messages by the use of a sending and receiving host-id and a sequence number of the packet that is being ACK'ed. In this model however, ACKs are controlled by a simple counter because the IP packet representation doesn't actually contain TCP/IP data.
- CSU/DSU Load 1/2/3/4/5 and CSU/DSU Failure Length 1/2/3/4/5: in the heterogeneous environment of the internet, where there is variance in the performance of different vendor's equipment, it has been observed [13] that these units can malfunction under

heavy loads. This behavior is modeled by setting these parameters. They are instantiated at the node model level because nodes can be comprised of different vendor equipment. Also continuing the thread from (Time to Reconfigure Network) above, it is less likely that a node at the upper layers of the hierarchy would have a malfunctioning CSU/DSU. Setting these parameters at the nodal level will allow for the fullest control of the simulation environment. While this behavior is not believed to be a large contributor to the present internet instabilities described in [13], the authors included it because a number of respected internet engineers have voiced that malfunctioning CSU/DSUs are part of the problem of anomalous protocol behavior. The source for the CSU/DSU information in [13] states that typical bit error rates are approximately 10e-5 to 10e-6 [1]. He wasn't aware however about the percentage of bad CSU/DSUs deployed. But according to [13; 22], the actual damage introduced into the fabric as a result of the CSU/DSU anomalies is minimal compared to specific vendor implementations (i.e., operational shortcomings) of BGP. So the representation of the malfunctioning CSU/DSU behavior in this model will be minimal.

The bit errors are modeled by a delay value (CSU/DSU Failure Length 1/../5) which is designed to mimic the delays induced by requests for retransmissions of packets received with bit error > 1. It is assumed that one bit error in a packet can be forward corrected whereas >1 bit error will cause a request for retransmission. As traffic rates are increased to saturation points, the bit error rates, and associated delay times used in the model will increase. The nominal case is derived using bit error rates of (10e-5.5). Corresponding delay times are related to the probability of packet error where there is more than one bit in error. In this model, an estimation is made that at operational levels greater than or equal to 76% *LC* in the CSU/DSU, the probability of >1 bits in error in a packet is equal to the probability of a one bit error in a packet while the CSU/DSU is running under the 76% *LC* level. The probabilities increase as the load increases (which are also estimations) shown by the following table. In this model, traffic is not delayed to model bit errors while link operational capacity does not exceed 76%, therefore a zero probability is shown as the first entry. The probability that a packet contains a bit error is:

$$PPE = BEP * MPS * 100 \qquad (4)$$

where

PPE = Probability of Packet Error (%)

BEP = Bit Error Probability (given above as 10e-5.5)

MPS = Mean Packet Size (bits)

Table 5. CSU/DSU Load Versus Packet Bit Error Probability

Load (%)	**BEP >1/Pkt (%)**
1 - 75	0
76 - 80	0.708
81 - 85	0.850
86 - 90	0.992
91 - 95	1.417
96 - 100	2.833

Packet delays are derived from Eqs (1) and (4). Since any packet crossing the link while the load is at or above the threshold has a *PPE*, a uniform delay value could be applied at a steady rate to mimic retransmission delays. The values are derived by the following equation and shown in Table 6.

$$CSU\text{-}D = (PPE/100) * 2(TV\text{-}AVG) \qquad (5)$$

Where

$CSU\text{-}D$ = The delay experienced by packets traversing the CSU/DSU module (seconds)

TV-AVG = The average of all link's timeout values (= 0.192 seconds)

Table 6. CSU/DSU Load Versus Packet Delay Time

Load %	Pkt Delay (Seconds)
1 - 75	0
76 - 80	0.0027
81 - 85	0.0033
86 - 90	0.0038
91 - 95	0.0054
96 - 100	0.0109

- Max Hop Count: the packet hop count is implemented by one each time it traverses a node on it way to the destination. The hop count value is initialized to zero for all packets. This parameter is used in the internet today to kill packets that may be in a transient routing loop or experiencing other pathological routing behavior. In the model this parameter can be accessed to indicate that the network may be undergoing a route flapping storm. That is, from a protocol convergence standpoint, the routes are changing more quickly than the time it takes the protocol to compute/promulgate consistent reachability information. This means that packets are not being correctly routed to destination. In this model only BGP nodes can recompute the network because there is an assumption of correct operation of intra-AS protocols. When the network is being recomputed however, only the packets at the node that is recomputing the network are delayed during that operation. It is conceivable that if several BGP nodes were recomputing the network closely together in time, then packets would start to be killed because they exceeded their maximum hop count. In this model the maximum hop count is set to be approximately equal to the maximum diameter plus the mean diameter. This will ensure that a packet can undergo the worst case hop count plus have a margin if the packet were re-routed in transit. The maximum diameter of this model is 11 which is nearly equivalent to the observed diameter of the inter-domain topology in [7:11]. The mean hop based diameter is derived from a matrix representation of the network where the entry H(i,j) is the minimum number of hops that is takes to get from node i to j. This hop count is not a "routed" hop count. That is, it does not consider minimum link cost, but only the existence of the link. Therefore it is a best case estimate on number of hops that any packet will take from source to destination. The mean hop based diameter is obtained by summing all the elements of the matrix and dividing by the number of entries. My model has a mean hop based diameter of approximately 4.3042. The Max Hop Count parameter in this model will initially be set to 15. The goal is to have less than 0.015% of the total packets lost due to hop count exceeded under nominal loading. This percentage corresponds to measurements taken on the internet using traceroutes between 37 sites [21].

- BGP Timeout: this parameter is 90 seconds which is, by specification, equal to three times the BGP Keepalive Time which in common practice is 30 seconds [22].
- BGPK Processing Delay: the BGP keepalive (BGPK) message is represented by only a BGP header and carries no data payload and only serves to reset timers. In this model, data packets are not delayed because a BGP node is receiving a BGPK message. The delay value itself is set to 0.001 seconds. The time that it takes a router to process a BGPK is an unknown quantity and the delay is an estimate. But the BGPK message receives minimal processing by the receiving router, (the sending host ID is checked to verify that it parses the local access list), and at only 152 bits, about 1000 machine cycles doesn't seem an unreasonable resource allocation to process the message.
- BGPI Processing Delay: a BGP interim update (BGPI) message can contain one route announcement or multiple route withdrawals [8:116]. The BGPI Processing Delay will be instantiated as a fraction of the amount of time it takes a node to recompute the network. The value of this parameter will vary based on the position of the node in the

hierarchy. For instance, the top level nodes will receive on average more route withdrawals than announcements. This means that the BGPI Processing Delay value will be proportionally larger there than at some of the other nodes that are lower in the hierarchy.

A baseline value for the BGPI Processing Delay will assume that route withdrawals and announcements are roughly equivalent over time. However, once established, the baseline will be increased at nodes in which flapping behavior is being modeled (many more withdrawals than announcements). The value is given by

$$BGPIPD = TRN * (1/NE) * (ERW/ERA) \qquad (6)$$

where

$BGPIPD$ = BGPI Processing Delay (seconds)

TRN = Time To Reconfigure Network for the node in question (seconds)

NE = Number of Edges in the Transit-Stub ISM (= 78)

ERW = Estimated number of Route Withdrawal messages

ERA = Estimated number of Route Announcement messages

The number of edges refers to the connectivity of the Transit-Stub Internet Survivability Model and is used in the computation because a node could receive a BGP message containing information on any of the 78 links in the model. The estimated route withdrawals to announcements is a ratio that indicates the amount of instability that any particular node may be expected to experience. Since the model uses an abstraction of the BGP protocol, this ratio can be arbitrarily chosen but it is based on the findings in [13]. Likewise, the number of nodes that are experiencing redundant BGP route withdrawal updates is small as indicated in [13]. ERW/ERA is normally expected to be 1 or very close to it.

- Mean Link Down To Up Delay: the Cisco 7500 series router mentioned in the Time to Reconfigure Network section above and being modeled here in the top layer of the hierarchy, re-establishes a lost BGP peering session within about 15 seconds [20]. That value will be used between nodes 1, 2, 3, and 4 in this model. The lower layer BGP nodes can be representative of a Cisco 2500 series router. Reconnect times in these routers have been observed to be about 90 seconds [20]. Because any potential BGP peer can initiate the BGP session, the lesser of the reconnect times will be used in situations where a larger border router is peering with a smaller one in this model.
- Maximum / Mean BGP Packet Length: since this model abstracts actual BGP messages by treating packets as messages, the Mean and Max lengths will be the same as the mean and max lengths for regular network packets which is 1120 bits and 12000 bits respectively. This convenience for the sake of modeling does not represent a large divergence from the accuracy of the model representation, however, as the key factor in this case is not the transmission delay based on packet size, but the amount of processing the BGP messages cause. The 1120 bit figure was observed as the mean packet size for IP packets on the NIPRNet [3] and the 12000 bit figure is the maximum IP packet size allowed. The packet lengths are generated by a exponential random number generator which is provided the mean length as its parameter.
- BGPIK Traffic Proportion: this parameter is the BGPI to BGPK message proportion. It is not widely known, but can be estimated as a function of overall route stability on the internet. However, this estimation is not a tight correlation because each border router still must process all BGP updates with respect to its local routing policies and route preferences. The Transit-Stub ISM simulations will have a duration of roughly 10's of minutes. A corresponding route stability figure for that time period was measured in the internet in [21]. The measurements suggest that when considering stable routes, (routes

where no routing pathologies such as persistent/temporary loops, erroneous routing, mid-stream routing change, or lost packets were observed), overall stability is about 91%. This model will used that distribution as a estimate of BGPIK Traffic Proportion. The figure is actually used as an input to a cumulative distribution function random number generator where the probability of generating BGPK messages is 91% and a corresponding 9% for BGPI messages.

- Total Network BGP Traffic: this figure has been observed in the internet to be much less than 1% of the Total Network Traffic [14]. It is instantiated here as (.005 * Total Network Traffic) as a baseline but is dependent on the node. The amount of BGP traffic will actually be dependent on the Total Network Traffic for the nominal case simulation which will be reported in chapter 5. In my model, some nodes will have more BGP information to pass than others. See Appendix A for a node by node explanation of the amount of BGP traffic at the 14 BGP nodes.
- Capacity A/B/.../J: These values are link capacities in bits/sec and were chosen to make the simulation times reasonable.
- Node Out Degree: The instantiation of this parameter is self explanatory but its use as part of the "Processing BGP I?" and "Recomputing Network" modules (see Figures 41 and 42) is explained here. During the times that a router is processing a BGP interim update message or is recomputing the full network (i.e., its maximum knowledge of the network), packets are still being switched [20]. However there are different levels of switching on a router. If the route is part of cache, then the packet is switched almost instantaneously on the backplane of the router (in firmware) where no CPU interaction is required. When the network, or a portion of the network is being recomputed, the probability that the next hop information for any given packet will not be in the cache is increased. Therefore the overall node traversal time for packets increase. This delay in

traversal time is modeled here as a percentage of the overall time that it takes to process the update message. Packets are delayed by the used of an Absolute Delay block. The delay is set to

$$BGPIPD * (1/NOD) \tag{7}$$

where

NOD = Node Out Degree

for BGPI messages, or

$$TRN * (1/NOD) \tag{8}$$

for BGP full update messages. The inverse of the node out degree is used because the BGP update may be associated with any of the links entering the current node.

When the Transit-Stub ISM is being simulated where BGP messages are being hashed, the packet traversal times will double for (7) and (8). The same principal holds for packet delay probabilities as when no hashing is being used. That is, if the route is available in cache, the packet is switched on the backplane with no CPU involvement. However, when the CPU is called upon to make a routing decision when hashing is being modeled, the CPU resource will be less available to compute the next hop of the requesting packet. Since BGP is run without hashing in the real internet [14], no data is available that gives an indication as to how packets may be affected if hashing were used. The 2x slowdown value is an estimate.

The MD5 hashing algorithm performance was measured on a Sun Sparc Station 4 running Solaris 2.5 [10]. The time that it took to hash a message with the mean packet size that is being used in the Transit-Stub ISM, (1120 bits), was 0.00004 seconds. This doesn't not count comparing the hash values between the received message and the one just computed. It also does not take into account that the router's processor will not be dedicated to performing only hashing. That notwithstanding, the time that it takes to compute a hash is considerably less than the time it takes to process a BGP interim or full update message. But all BGP messages are hashed, including Keepalive messages. The number of Keepalive messages in the Transit-Stub ISM can be expected to be 10 times greater than the number of update messages. That is why that, despite the fact that hashing is less overhead for the router than processing a BGPI/F message, the packet delay time is considered equivalent over both in this model. Still the 2x estimation of packet delay is a worst case scenario. Even using the factor of ten found in the BGPK:BGPI/F message type ratio and applying it to the time it takes to compute a hash, the value of 0.0004 is still much less than the time required of a router to update its tables based on the content of a BGP update message.

Basic Model Operation

The model operation as described here is top down and meant to give an overall view of the course a packet may travel through the network and the modules acting on that packet. Before data traffic is generated and the network undergoes a routing simulation, the Init Network module executes. The initialization process reads the link cost information file and loads it into the Cost Matrix memory. This Cost Matrix is used as input to the Compute Routing Matrix module. The Compute Routing Matrix module uses the Dijkstra algorithm to compute the least cost paths in the network and set the topology [19]. The Traffic Matrix, which is used to shape the end-to-end traffic on the network, is also created. All matrices are square N x N, where N = the number of nodes in the network. An entry N_{ij} represents the a node in the network with the row index used as the from node and the column index, the to node. The Init Network module is also responsible for keeping a running average of the composite end-to-end delay in the network. After the initialization process is complete then traffic generation starts. Since there are different types of traffic being produced in the network, each is given separate treatment here. Whether or not the type of traffic is data or BGP, it is represented by the BONeS data structure the WAN Packet. The following table gives its description.

Table 8. WAN Packet Description

Name	Type	Subrange	Default Value
Source Host	Integer	[1, +Infinity)	1
Destination Host	Integer	[1, +Infinity)	1
Source IMP	Integer	[1, +Infinity)	1
Destination IMP	Integer	[1, +Infinity)	1
Tx Start Time	Real	[0, +Infinity)	0.0
Hop Count	Integer	(-Infinity, +Infinity)	0
Length	Integer	[0, +Infinity)	...
Type	Packet Type Set	...	Data
Status	Status Set	...	OK
Time Created	Real	[0, +Infinity)	0.0
Data	Root Object	...	...

Data Traffic (including acknowledgment packets). Each node generates data traffic from the WAN Traffic Gen module. The traffic generation module is located in the Node module and is representative of the session (and above) layers of the OSI model. The traffic generation is a memoryless, or Poisson, process as shown by Eq (9) below. Traffic may be destined for any node in the network. The traffic interarrival times in packets per second are based on three values, (Traffic Matrix, Traffic Matrix Sum, and Total Network Traffic), and its memoryless property is induced by the use of a exponential random number generator. The traffic destination rate is controlled on a per-node basis by the Traffic Matrix. The index (i,j) is a number that represents a relative amount of traffic (in packets per second) generated at node i and bound for node j. The Traffic Matrix Sum is the sum of all entries in the Traffic Matrix. The Total Network Traffic is instantiated at the simulation system level as a number representing packets/sec. The traffic rate is computed by the following

$$TR(i,j) = (\cong 1)/[(TM(i,j)/TMS) * TNT] \tag{9}$$

where

TR(i,j) = the Traffic Rate between nodes i and j (packets/≅second)

TM(i,j) = the Traffic Matrix value in row i and column j (packets)

TMS = the Traffic Matrix Sum (packets)

TNT = the Total Network Traffic (packets)

≅1 = the value generated by an exponential random number generator with a supplied mean of 1

The traffic pattern throughout the network is nearly uniform, but a given node is more likely to generate inter-AS traffic, than traffic destined for it own AS.

Once the traffic leaves the traffic generation module, it enters the network module. The behavior of the a data packet is different between the BGP/WAN Node (which represents an AS border router), and the WAN Node. Within the WAN node a packet may enter from two points. Either it has just been generated at this node and is entering from the transport layer or it is coming from another node in the network. In the former case, the packet goes through logic to route the packet to the next hop towards the destination. This is done with the Lookup Next Hop module that accesses the Routing Matrix memory. The Destination IMP (interim message processor) node is inserted and the packet leaves the node via the data link layer. If the packet is incoming to the network layer of the node via the data link layer of another node, then the packet has its hop count incremented. The hop count value is then compared to the Maximum Hop Count parameter and the packet is sunk and written to a global memory structure (for reporting purposes) if that value is exceeded. If the hop count is an allowable value, then the packet is either delivered to the transport layer if the current node id is equal to the Destination Host field of the packet, or the packet enters the routing logic blocks where its next hop is determined. If the packet is destined for the current node, then at the transport layer the packet is measured for how long it has been in the network (TNOW - Time Created) and this value is used to update the end-to-end delay global memory. The packet is then retired.

Before the data link module is described, the extra actions that a packet may undergo at BGP/WAN node are described. The BGP/WAN Node is the only nodal module that can accept BGP messages. As such, it is the only nodal module that can recompute the network on either a full or partial basis. If the current node is a BGP/WAN node and it is currently processing either a full or partial BGP update, then a Yes/No memory switch is enabled. If yes, then packets that traverse this node during the update period are routed through an Absolute Delay module to be delayed for a fraction of the time that it takes the node to process the BGP update.

Once at the data link layer (DLL), the data packet undergoes several operations. A packet may enter the DLL from the current node's network layer of from the physical link attached to a remote node. If the packet is entering from the network layer, it first is routed through a Timestamp module that inserts the time of transmission from the network layer above. This is used to keep track of timeout values. The data packet also gets updated by having the current node ID written to the Source IMP field. This information is used at the next node to make a routing decision. A copy of the packet is then delivered to a hold buffer where it is held until either an acknowledgment is received from the next node or a timer expire event occurs. The timer value is equal to *TV* (see Eq 1). If the timer expires before an acknowledgment is received, then the packet is retransmitted from the hold buffer where it reenters the Timestamp Module. If the acknowledgment packet is received before a timer expire event occurs then the duplicated packet in the hold buffer is discarded. The one-deep retransmission capability in this model is not like the sliding window protocols employed with TCP, but the extra traffic induced by retransmissions is meant to be an approximation to overhead induced by TCP.

After the duplicate packet is written to the hold buffer the original packet is delivered to a FIFO queue with priority. It is released from the queue after the packet(s) ahead of it undergo(s) a transmission delay given by (Eq 2). After the packet leaves the transmission delay module the packet timer is started. Once the timer has been set then the packet will enter a CSU/DSU

Behavior module if the current simulation time has progressed enough to let transient simulation startup behavior die out, the value is 10 simulation seconds. While in the CSU/DSU Behavior module the packet may undergo a delay based on current link utilization. This model is meant to mimic certain faulty CSU/DSU modules found in the internet today (as explained in the CSU/DSU Load 1/2/3/4/5 and CSU/DSU Failure Length 1/2/3/4/5 sections above). If the link utilization is below a user-defined threshold, the packet undergoes no delay in this module and enters the physical Link. At the physical link, the packet undergoes a propagation delay equal to the value given by Eq 3.

When a packet enters the DLL from the physical link on its way to the network layer, it will first enter the CSU/DSU Behavior module if the simulation time is greater than or equal to 10 seconds. After exiting this module it enters an ACK/Data switch. If the packet is an ACK packet it is routed to the Cancel Packet Timer Module that was set as the associated data packet left the DLL in the opposite direction. It is then routed to the hold buffer where it trips a trigger port causing the copy of the associated data packet to be sunk. Alternately, if no ACK packet is received before the timer expires, then the Service Packet Timer block will execute and cause the data packet in the hold buffer to be retransmitted. If the packet entering the DLL from the physical link is a data packet then it causes an ACK packet to be transmitted in the opposite direction toward the sending node before exiting the DLL towards the network layer module. The ACK packet is sent into the data stream going in the opposite direction just before a Packet Priority module. The ACK packet is assigned priority and then sent into the FIFO queue with priority where it assumes the position at the head of the queue. The priority discipline is no preemption, so if a packet is being transmitted as the ACK enters the queue, then it assumes a position directly behind that packet.

BGP Traffic. BGP protocol traffic is produced only at BGP nodes and is only destined for those nodes. In this model the BGP node IDs have to start at 1 and be consecutively numbered.

This is because the Destination Host and Destination IMP fields of the BGP protocol traffic are produced with a 1 to N do loop (the data traffic destination fields are filled out in the same way (see Figure 20)). The N value in the 1 to N is the Number of BGP Nodes simulation parameter.

The BGP protocol process belongs to the network layer and the traffic is produced in the BGP PDU module which is unique to WAN/BGP Nodes. The traffic is produced in the same way that data traffic is produced. But because BGP traffic is based on immediate neighbor acquisition to set up peering sessions, the traffic is not routed but injected directly into the DLL where it undergoes the same processes as does the data traffic.

The differences in the processing of BGP traffic are all at the BGP/WAN Network Layer. The outgoing BGP traffic is handled by the BGP Out Processor. This is a simple extension to the BGP traffic generator. The only logic the outgoing packets pass through is a session tester. If the remote peering session with the Destination Host is down, then the packet is sunk, otherwise it is transmitted.

The incoming BGP traffic is filtered through a BGP switch at the network layer and sent to the BGP PDU. Within the BGP PDU module the traffic is split and processed based on the sending Node Number. Each stream of traffic from any of the allowed neighbors, (BGP peers (which are the one-hop BGP neighbors of the current BGP node)), undergo identical processes in parallel. The traffic is first sent to the BGP Memory Test module. If the current peering session with the sending peer is down then the incoming traffic is sunk unless it is control traffic in which case it is used as session negotiation (see Appendix A, Figures 36, 37, and 40, and accompanying text, for the full explanation of the session negotiation process).

If the BGP traffic is not control traffic, and if the peering session is active, the packets are passed from the BGP Memory Test module to the BGP In Processor module. Once in the BGP In Processor, the packets control timer events and are input to a processing delay block which mimic the delays associated with processing BGP messages. If the packet is the first one seen by this module after simulation start, or after the re-establishment of a failed BGP peering session then a timer is started. Each subsequent packet first cancels the active timer and then immediately restarts it. This allows for control of the BGP PDU based on interarrival times of BGP messages. Any BGP message is sufficient to reset the timer as is the case with the actual operation of the protocol in the real internet. If the packet is the first one seen by the receiving peer from the particular sending peer after a failed BGP peering session is re-established, then the packet is processed as a BGP full update (BGPF) message. Otherwise the packets are a mixture of BGP Keepalive (BGPK) messages and BGP Interim Update (BGPI) messages. The mixture function is given in the BGPIK Traffic Proportion section above.

Within the BGP In Processor the packet receives simulation processing time slices based on the type of packet it is. The BGP updates are all tied to a simulation resource parameter: Resource For Processing Delays. The contention for the allocation of the CPU is modeled by a FIFO queuing processes without priority or preemption which is set up as a dedicated server. While the CPU process is active, a memory parameter is set, (either the Processing BGP I? or Recomputing Network?), that indicates whether or not the node is undergoing an update process. If it is, then all packets traversing that node are delayed as shown by Eqs 7 and 8 above. After the packet receives the appropriate CPU time allocation then it is sunk.

If a timer expire event occurs because a BGP packet has not been received within the window, which is set to 90 seconds, then the associated session is terminated and the network is recomputed indicating that the peer is no longer available. This uses the same Resource For Processing Delays as explained above. After a delay that is the result of a normal random number generator that outputs a value based on the Mean Link Down to Up Delay parameter with a variance of 2 seconds, the session is reestablished and the network again recomputed indicating that the failed node is back online.

Baselining the Model, Running Simulations, and Gathering Data

To arrive at nominal model operation when running a simulation, certain thresholds have to be defined and met before the model can be declared stable. Also, baselines have to be established against which subsequent simulation data can be compared so that meaning can be given to data that varies from the baseline. In this case there are a few factors that will help define the baseline. The traffic rate will be configured to load the links that fill most quickly at an average of 25% utilization. During peak traffic hours, the average utilization jumps to about 50%. These percentages were defined by one ISP owner as the value to shoot for when filling up available bandwidth [2].

The percentage of packets lost due to hop count being exceeded should also be within the 0.015% upper limit as given in the Max Hop Count section above. Also, the queue mechanism at each link should reject less than 0.1% of the total traffic across that link during the simulation. This 0.1% figure is an estimation on my part and will be used as a model tuning parameter. That is, if more than that amount of traffic is rejected by the queue during the nominal simulation, then the queue size will be increased in the model. BGP timeouts, if any, should be rare and no cascading effect should be observed where, when one node recomputes the network, then that leads to further congestion and recomputations of the network by its BGP peers.

Traffic shaping was done based on a representation of the changing traffic patterns evidenced in the internet with the advent of WWW servers. Within this model, a node is twice as likely to generate inter-AS traffic than generate a packet destined for a node in its own domain. This traffic shaping is controlled by the Traffic Matrix memory. The actual traffic generation is modeled as a memoryless exponential process as is standard in network simulations. That is, traffic at each node is generated independently of traffic at any of the other nodes and traffic interarrival times are independent events. The computation is made on a mean packet size of 1120 bits. This mean is observed on the CONUS NIPRNet administered by DISA

[3]. With these baselining figures as a starting point, the model can be adjusted (network tuning) to support the nominal case. The tuning of the model would include increasing queue sizes and link capacities where appropriate.

The Transit-Stub ISM contains several random number generators that are used to control traffic rates and times between events. Within the BONeS Designer environment, at the time that any one generator is used, it is supplied with a unique large integer to act as seed. However, if the number supplied to each generator is -1, then the global random number seed is used. In this manner, identical simulations can be run until the model is baselined. Then different global seeds can be supplied for different simulations of the model. This capability will allow for a more robust verification of the model. The random number generators in the Transit-Stub ISM all use -1 as their seed allowing the global seed to control the simulation.

Once all this information is obtained, the model can be used as a guide to survivability in the fabric. The main control parameters will be traffic load, a representation of CSU/DSU malfunction, priority assigned to protocol messages, and the use of hashing of BGP messages. Once the simulations are run, various factors can be observed to glean the current survivability picture: traffic loads compared to number of BGP failures, traffic loads compared to non-received acknowledgments, the states of queues in the model as the survivability of the fabric is modeled, the average end-to-end delay of packets in the network, the distribution of BGP session failures over time and over traffic loads, and other pertinent model behavior representations.

After the general survivability information is gathered, the model can then be extended to represent a system where the control traffic is protected via MD5 hashing. Contextually identical simulations can be run and the fabric re-evaluated for survivability in the context of the introduction of specific measures designed to guarantee authentication and integrity. This finally gives an indication of the advisability of employing the mechanisms in the current internet to defend against DoS.

Finally, a current characterization of our vulnerability to DoS attacks can be given that will allow for the formulation of risk analysis data that can lead to a decision about the level of survivability protection to engineer into the current/near future internet. The actual risk analysis is outside the scope of this work largely because a representation of the current threat is not easily quantified. But it may suffice to point out that current internet control traffic is not safeguarded against hacking and that DoS attacks are on the rise as seen by the CERT and are the easiest type of attacks to mount and the hardest to defend against [17].

Summary

The process of accurately representing the problem domain will leave areas for refinement that will be addressed in chapter six. However, assuming for the moment that the model and its parameters are a fair representation of the domain, this still leaves the task of tuning the network so that its links and queues are sized to the amount of traffic introduced over them. This will mean that the network as represented in Figure 4 may change slightly as the model is being baselined. The first part of chapter five will explain those refinements as well as the process of tuning the network. Once that is done, then the information gained through simulation will be as pertinent as is possible given the accuracy of the model's domain representation.

V. Simulations and Analysis

Introduction

This chapter describes the modifications made to the model so that its simulation run times could be reduced from over 10 days per simulation to around 24 hours. The model reduction is made by discarding all but the BGP nodes which results in a 14-node model. This was the logical collapsing point between the two models, as the research is based on the assumed correct operation of intra-AS traffic. Simulation results of the nominal run of the scaled-down model are compared to nominal simulation run of the 50-node model and found to be a close fit after adjustments are made to the link sizes of the former. The queue sizes of both models are found to be adequate under nominal loading conditions. The queue sizes remain adequate until intentional link flooding is simulated.

After the baseline 14-node model is tuned to emulated the behavior of the 50-node model, it is put through a series of simulations in which the independent variables of, traffic rate, prioritizing BGP traffic, and hashing BGP traffic, are controlled. The results are then discussed and conclusions drawn based on the values of the key dependent variables of network end-to-end delay, packet hop count, and BGP traffic delay.

Model Scaledown

The performance of the 50-node simulation on BONeS Designer was found to be very slow. For the nominal case run where the links that filled most quickly were injected with enough traffic to produce an approximate 25% steady-state utilization rate, the amount of network traffic that had to be simulated was 400 packets/second. A 200 second simulation took 948 minutes to complete. The corresponding case for the 14-node simulation, which requires a total network traffic value of 150 packets/second to reach nominal performance, takes just 60 minutes to complete. However, when the traffic rates increase, the simulation times appear to grow in an exponential fashion. For example, the 14-node simulation, using a traffic rate of 600 packets/second, took 18 hours to complete. The rough extrapolation to the 50-node simulation using a Total Network Traffic value of 1600 packets/second indicates an approximate 12 day completion time. Since sixteen simulations have to be run per single seed value (see the Simulation section below), the 50-node simulations are too lengthy to provide results in a reasonable amount of time.

The solution is to try to scale the model down by collapsing the network into a representative subset of the original model. A logical breaking point in this model is to use the 14 BGP nodes only (the model's BGP nodes are numbered 1 through 14 in Figure 4). This is because the non-BGP nodes comprise the intra-AS network and are given minimal representation in the 50-node model. The model can be collapsed along the BGP nodes if they can be configured show similar simulation results to the 50-node model. In the Transit-Stub ISM, the parameters of the model allow an independent representation of BGP policy (and associated processing delays) on a nodal basis. This allows for flexibility in the nodal representation and permits a reasonable collapse.

In order to test the validity of scaling the simulation system down, the 14- and 50-node nominal traffic rate simulations will be compared using end-to-end delay of all data packets on the network, the average hop count of all the packets on the network, and link utilizations of the links between the 14 BGP nodes. Each system will be simulated for 200 seconds. Since the two systems will be compared using a nominally loaded network, link queuing delays will not be a factor as the transmission queues will not have packets accumulate in them more quickly than they can be transmitted.

In order to ensure that the comparison of the two simulations is fair, the amounts of traffic were computed to establish an approximate 25% link utilization rate for the links that fill up most quickly. The same traffic matrix is used for the core 14 nodes between the models, and a symmetrical traffic pattern was used between the top level nodes (nodes 1 through 4) and the lower level nodes. In the case of the 14-node simulation, the amount of traffic transmitted to and from the top-level nodes and the remaining 10 nodes was 1.98:1. In the 50-node simulation, the amount of traffic transmitted to and from the top-level nodes and the remaining 46 nodes was 1.95:1. The following figure provides detail about how the traffic pattern is established in the model. Recall that Eq. 9 gives the formula for the computation of the traffic rates in the model.

Traffic Pattern: 14 Node Model

Relative Traffic

Node Frm/To	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	
1	0	6	6	6	4	4	2	2	1	3	2	1	1	1	Traffic Matrix Sum
2	6	0	6	6	1	1	3	4	4	1	2	1	1	1	354
3	6	6	0	6	1	1	2	1	7	4	4	4	1	1	Total Network Traffic
4	6	6	6	0	1	1	1	2	1	2	1	5	4	4	150
5	7	1	1	1	0	1	1	2	1	1	1	1	2	1	
6	4	1	1	1	2	0	1	1	1	1	1	1	2	1	
7	2	4	1	1	1	1	0	4	3	1	1	1	1	1	
8	1	5	2	1	1	1	4	0	2	1	1	1	1	1	
9	1	3	3	1	1	1	2	2	0	1	1	1	1	1	
10	1	1	3	1	1	1	1	1	1	0	2	2	1	1	
11	1	1	3	1	1	1	1	1	1	2	0	2	1	1	
12	1	1	4	5	1	1	1	1	1	2	2	0	1	1	
13	1	2	1	4	1	1	1	1	1	1	1	1	0	2	
14	1	1	1	5	1	1	1	1	1	1	1	1	2	0	

Packets/Second

															Total (From Node)	
1	0	2.542	2.542	2.542	1.695	1.695	0.847	0.847	0.424	1.271	0.847	0.424	0.424	0.424	16.53	Avg Level 1
2	2.542	0	2.542	2.542	0.424	0.424	1.271	1.695	1.695	0.424	0.847	0.424	0.424	0.424	15.68	16.58
3	2.542	2.542	0	2.542	0.424	0.424	0.847	0.424	2.966	1.695	1.695	1.695	0.424	0.424	18.64	Avg Level 2
4	2.542	2.542	2.542	0	0.424	0.424	0.424	0.847	0.424	0.847	0.424	2.119	1.695	1.695	16.95	8.37
5	2.966	0.424	0.424	0.424	0	0.424	0.424	0.847	0.424	0.424	0.424	0.424	0.847	0.424	8.90	Ratio L1:L2
6	1.695	0.424	0.424	0.424	0.847	0	0.424	0.424	0.424	0.424	0.424	0.424	0.847	0.424	7.63	1.9810127
7	0.847	1.695	0.424	0.424	0.424	0.424	0	1.695	1.271	0.424	0.424	0.424	0.424	0.424	9.32	
8	0.424	2.119	0.847	0.424	0.424	0.424	1.695	0	0.847	0.424	0.424	0.424	0.424	0.424	9.32	
9	0.424	1.271	1.271	0.424	0.424	0.424	0.847	0.847	0	0.424	0.424	0.424	0.424	0.424	8.05	
10	0.424	0.424	1.271	0.424	0.424	0.424	0.424	0.424	0.424	0	0.847	0.847	0.424	0.424	7.20	
11	0.424	0.424	1.271	0.424	0.424	0.424	0.424	0.424	0.424	0.847	0	0.847	0.424	0.424	7.20	
12	0.424	0.424	1.695	2.119	0.424	0.424	0.424	0.424	0.424	0.847	0.847	0	0.424	0.424	9.32	
13	0.424	0.847	0.424	1.695	0.424	0.424	0.424	0.424	0.424	0.424	0.424	0.424	0	0.847	7.63	
14	0.424	0.424	0.424	2.119	0.424	0.424	0.424	0.424	0.424	0.424	0.424	0.424	0.847	0	7.63	
Total (To Node)	16.10	16.10	16.10	16.53	7.20	6.78	8.90	9.75	10.59	8.90	8.47	9.32	8.05	7.20		

Figure 5. Instantiation of the Traffic Pattern in the 14-Node Model

The top matrix is the relative amount of traffic between any two nodes in the network. The row index is the from-node and the column index is the to-node. The relative traffic value is divided by the traffic matrix sum and then multiplied by the total network traffic. The bottom matrix entries are the result of those computations and are representative of packets/≅second (see Eq 9). The figures at the end of the rows in the bottom matrix represent the total amount of traffic leaving the corresponding node. The figures at the bottom represent the total amount of traffic entering the node. The level 1 nodes are 1 through 4. The level 2 nodes are 5 through 14. The Avg Level 1 figure represents the averages of the row and column totals for nodes 1 through 4. The Avg Level 2 figure represents the corresponding averages for the other nodes. The layout for the 50-node simulation is similar but space considerations prohibit its display. Note that the same 14 x 14 matrix is used within the 50 x 50 matrix. For the 50 x 50 matrix, the Avg Level 1 entry is 16.71, the Avg Level 2 entry is 10.94, the Avg Level 3 entry is 6.22, the Avg Level 2 / Level 3 entry is 8.58, and the ratio Level 1: (Level 2 / 3) entry is 1.95.

Comparison of The 14- and 50-Node Simulations

The data gathered from both simulations came from probes placed at identical points within the 14 BGP nodes. Since the simulations had nominal traffic rates, no delays were encountered in the CSU/DSU units, nor were abnormal delays encountered in the transmission queues at the data link layers. There were no dropped packets from the transmission queues during either simulation. There were also no BGP session failures in either simulation which is to be expected in a nominally loaded network. The two simulations were run using the same global seed.

The end to end delay values are computed using averages of the delay in all packets in the network over window period of 10 seconds with averages taken every two seconds (see Figure 18 for an explanation of the Compute Global Average module). The small window size will help ensure that rapidly oscillating delay values are not masked by a longer averaging period. The values observed in the 14- and 50-node baseline simulations are similar. The 14-node simulation produced a mean delay value of 0.223502 seconds from 99 windowed observations. The 50-node simulation produced a mean delay value of 0.215419 seconds from the same number of observations. There is about a 4% spread between the two values. The end to end delay value observed in the 50-node simulation can be expected to be slightly less because there are 36 non-BGP nodes that have no delay representation except transmission delay, and intra-AS traffic, by definition, does not cross a border router. This means that any packet delay associated with crossing a BGP node, while that node is currently processing a BGP update message, is circumvented.

The hop counts that packets accumulate in the nominal simulations can be expected to be close to the average hop count values in the network. They would increase only if the network were undergoing rapid topological changes, that could cause packets enroute, to be routed to more intermediate nodes than necessary. In the Transit-Stub ISM, this would be indicative of BGP session oscillation. Since there were no BGP sessions lost in either simulation, hop count data can be expected to remain stable. Unlike the end to end delay values that are influenced by many factors in the network, the average hop count values can be predicted in this model and that prediction can be used as a validity measure of correct operation. Variations in the analytical and empirical data can be attributed to the fact the analytical hop count estimations are based on a walk of the network where path metrics are not considered, and the empirical data is based on routed packets where path metrics are considered. Also, the traffic pattern is not completely deterministic because the interarrival times are the output of a Poisson traffic generation model.

The expected average hop count value in the 14-node network is given by

$$EHC\_14 = (AHC\_Top * T\_Top) + (AHC\_Btm * T\_Btm) \qquad (10)$$

where

AHC_Top= the average hop count to nodes 1 through 4 from all the other nodes in the network

T_Top = the percent of all traffic in the network destined for nodes 1 through 4

AHC_Btm= the average hop count to nodes 5 through 14 from all the other nodes in the network

T_Btm = the percent of all traffic in the network destined for nodes 5 through 14

The value is (1.660714 * .432) + (2.471429 * .568) = 2.1212. The observed empirical hop count average from the simulation was 2.0017 for a difference of 5.6%. It was derived from

1400 data samples (100 samples from each node in the network). The same windowing scheme and output rates were used for the hop count averages as was for the end to end delay.

Since only the 14 top level nodes in the 50-node simulation were probed for data, the expected average hop count value in the 50-node network can be approximated by

$$EHC\_50 = AHC\_Top14 * T\_Ratio\_Top14$$

(11)

where

AHC_Top14 = the average hop count to the 14 BGP nodes in the network (= 3.3829)

T_Ratio_Top14 = the traffic ratio of the top 14 nodes to the bottom 36 nodes (= 2.2242:1)

The expected hop count for the 50 node simulation is (3.3829 * .69) = 2.3342. The observed empirical hop count average from the simulation was 2.6756 for a closeness factor of 87 percent. This is not as good an estimation of the expected hop count compared to the 14-node network, but the estimation for the 14-node network model had more of a global knowledge picture than for the 50-node network. In the 50-node simulation, hop count data was only derived from the 14 BGP nodes in the network.

The final factor to consider when testing the scalability of the 50-node simulation to the 14-node simulation is link utilization. When gathering the data for the utilization percentages, the first 10-seconds of the simulation data is ignored to allow the model to obtain operating stability. In all cases, this allows the first utilization data point gathered to be within the normal distribution of the observed values. The utilizations can be expected to differ somewhat because the 50-node network model has intra-AS nodes behind each BGP node that are generating traffic on the network. The 14-node model has to generate traffic from the BGP nodes as if it were representing the same intra-AS structure as the 50-node model. The link utilization factors between the simulations were not close. This means that the link sizes in the 14-node model will have to be scaled to make the traffic picture look the same as the 50-node model. The following table gives the link utilization percentages between the two simulations.

Table 9. Comparison of Link Utilization Percentages

Link	14 Node Simulation (%)	50 Node Simulation (%)
1 → 5	3.68	17.38
1 → 6	6.33	8.57
2 → 7	4.31	20.21
2 → 8	4.12	11.47
3 → 9	4.05	22.24
3 → 12	6.79	14.79
4 → 12	5.08	14.19
4 → 13	3.25	15.25
4 → 14	6.25	15.83
5 → 6	1.77	1.76
7 → 8	0.04	0.03
8 → 9	0.77	0.84
10 → 11	3.64	12.76
11 → 12	6.83	18.64
13 → 14	1.84	2.14

Since the concern is the scalability of the model, the links in the 14-node simulation model will be made to be non-standard sizes based on the ratio of data in Table 9. The differences in link utilization percentages can be overlooked in the nominally loaded network as pathological behavior doesn't show up until the network becomes loaded with traffic. But since this research is concerned with the pathological case, the link utilization scalability factor will have to be commensurate between the two models.

After adjusting the link sizes in the 14-node model by factors commensurate with the link utilization ratios shown in Table 9, and re-running the simulation with the same global seed, the link utilization percentages matched those observed in the 50-node simulation to the 10E-4 level of precision. The hop count values remained as before, but the end to end delay went up slightly as is to be expected because smaller link capacities lead to longer transmission delays. The new delay value is 0.241892. This is 11% larger than the delay noted in the 50-node simulation.

However, a worse case delay value in the 14-node model will not distort the outcome of the research. It can be thought of as providing a more robust "no worse than" boundary.

The data between the two simulations suggest that the results obtained from the 14-node model will scale reliably to the 50-node model. A further test of the validity of the 14-node simulation is to compare its outputs to a simulation of longer duration. Due to the length of the simulation execution times, it is unreasonable to exercise the model to obtain days, or even hours worth of data. However, a 1000 second simulation with the nominal traffic rate of 150 packets/second takes a reasonable 18 hours to complete. The comparison of the 200- and 1000-second simulations are in Table 10. The same global seed value was used between the two simulations. Neither simulation had any lost BGP peering sessions, nor were any packets retransmitted because of timeouts. Also, there were no packets rejected from link transmission queues because of saturation. The maximum number of packets in the transmission queues were under 1% of capacity in all cases. Because network reachability data remained constant, link utilization averages did not oscillate wildly. There was no indication of model instability during the simulations.

Table 10. Comparison of the 200- and 1000-Seconds Simulations of the 14-Node Model

Simulation/Observation	**200-Second**	**1000-Second**
Average Delay (seconds)	0.2419	0.2688
Average Link Utilization (%)		
1 → 5	17.38	16.45
1 → 6	8.57	8.25
2 → 7	20.21	20.05
2 → 8	11.47	11.26
3 → 9	14.78	14.74
3 → 12	22.24	21.98
4 → 12	14.18	14.62
4 → 13	15.25	15.62
4 → 14	15.83	15.27
5 → 6	1.76	1.41
7 → 8	0.03	0.02
8 → 9	0.84	0.88

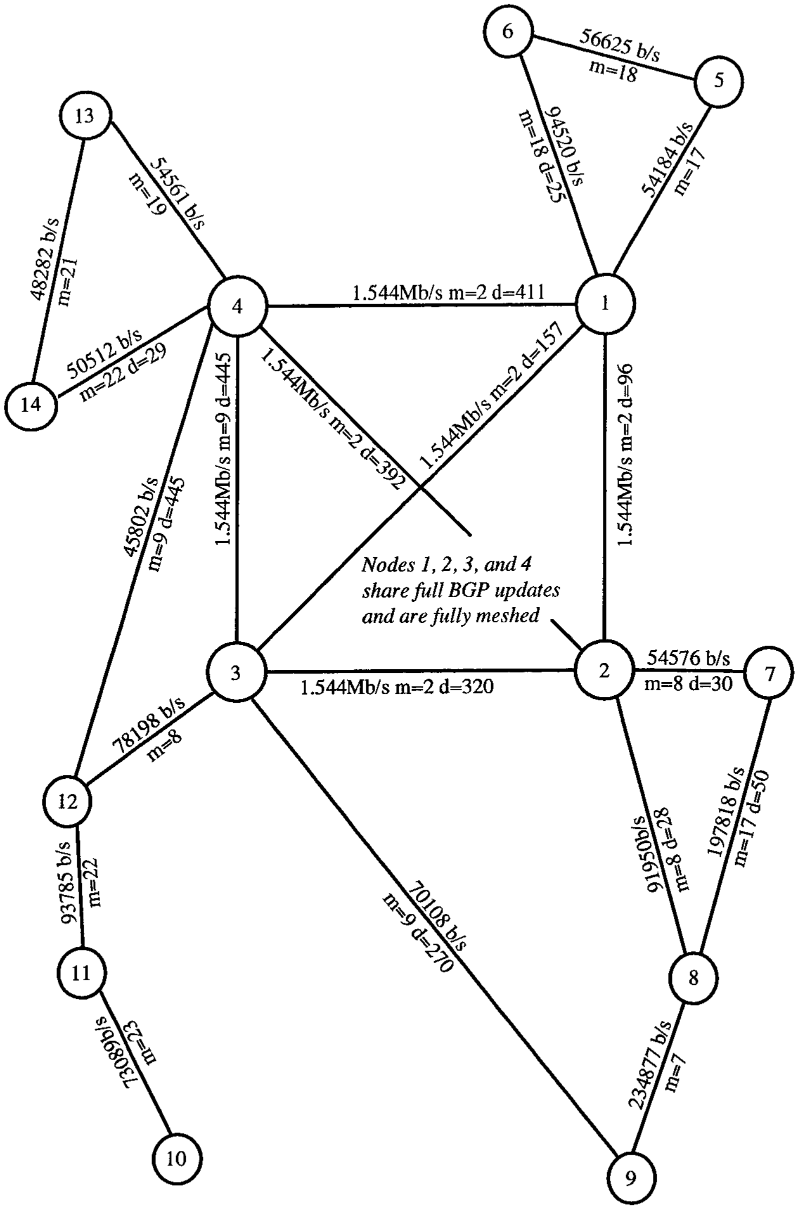
Simulation/Observation	200-Second	1000-Second
10 → 11	12.75	13.06
11 → 12	18.63	18.88
13 → 14	2.14	2.13
Average Hop Count	2.0017	2.0189

In addition to the similarity of the data, the hop count and delay data points in both the 200- and 1000-second simulations have a normal distribution and fall predominately within three standard deviations for the data samples, and in all cases fall within 4 standard deviations. The following table illustrates this.

Table 11. Distribution of Packet Delay and Hop Count Data

Distribution/ Simulation	Data Within 1 Std Dev (%)	Data Within 2 Std Devs (%)	Data Within 3 Std Devs (%)
200 Second Simulation			
Packet Delay	68.7	97	100
Hop Count	67.4	94.8	99.3
1000 Second Simulation			
Packet Delay	66.9	96.6	99.6
Hop Count	66.8	93.2	99.9

Due to the similarity of data between the simulations, the 14-node model, exercised for 200 seconds will provide reliable data for the research. The as-tested Transit Stub ISM is shown below. As in the 50-node Transit-Stub ISM shown in figure 4, *m* is the link metric and is also the distance of the link in miles unless there is a *d* entry, in which case, that entry is the distance in miles.



Nodes 1, 2, 3, and 4 share full BGP updates and are fully meshed

Figure 6. The 14-Node Transit Stub ISM

Simulation Results

The number of simulations required to exercise the control variables is 16. Four traffic rates are being modeled starting with the nominal case of 150 packets/second, and increasing by a factor of 2, 3, and 4. For each of these simulations, BGP traffic either will or will not be assigned transmission priority, and either will or will not have hashes computed. This is sufficient to cover the range of simulations given the use of one global seed. However, in order to establish validity in the model, the number simulation sets must be large enough to establish confidence that the observed behavior is correlated to the value of the control parameters. The actual number of trials run will be a function of how normally distributed the data is and the amount of confidence given the results.

The BGP traffic rate for all simulations is 10 packets/second (note that the BGP traffic generation is controlled by Eq 9 just as the normal data traffic is, so that the 10 packets/second figure should not be interpreted as absolute node-to-node traffic). This high number was picked to ensure that the output of the Poisson traffic generator would be such that there would be little chance of a BGP interarrival rate exceeding 90 seconds in long simulations. This value is the only best-case value assigned to any parameter in the model. However, choosing this high number gives a worst-case scenario to the delays experienced by data traffic in the network because the delay of data traffic traversing a BGP node is controlled by factors given in Eqs 7 and 8 where a higher rate of BGP traffic implies more average delay.

There are four model configurations that are simulated at varying traffic rates. The configurations are based on the factors of hashing and prioritizing BGP traffic. They are:

- Prioritize BGP Traffic = No; Hash BGP Traffic = No. This is the predominant operating environment in the live internet today.
- Prioritize BGP Traffic = No; Hash BGP Traffic = Yes. This is an unlikely scenario. There are currently efforts underway to encourage the use of priority for BGP traffic, but the idea of hashing to protect the authenticity and integrity of the BGP protocol traffic has not been widely encouraged/considered.
- Prioritize BGP Traffic = Yes; Hash BGP Traffic = No. This is the most likely near future scenario to be used in the live internet. The control of BGP peering is currently being instituted via access lists only [14].

- Prioritize BGP Traffic = Yes; Hash BGP Traffic = Yes. This represents the best of both worlds from a DoS perspective. From an efficiency perspective, the addition of hashing the BGP protocol traffic increases the end-to-end delay in the network. As will be shown by the following results, prioritizing BGP traffic ensures the viability of the topology even under pathological traffic loads, and the use of hashing ensures that the protocol messages will not fall prey to a DoS attack.

When BGP messages are assigned priority, they get moved to the front of the transmission queue in the Data Link Layer module (see Figure 24). The placement in the queue is done without preemption, (any packets currently being transmitted are not interrupted), and the packet to be prioritized gets placed behind any packets currently in the queue that have the same priority. In this model, acknowledgment packets have the level of priority equal to the priority assigned BGP traffic (see Figure 29). When BGP messages are hashed, the only dependent variable that can be affected is the network end-to-end delay. See the Node Out Degree subsection of the Instantiation of Particular Model Parameters section in Chapter 4 for an explanation of packet delay values associated with the use of hashing in the Transit-Stub ISM.

For ease of reference in the following discussion of simulation outcomes, the simulations will be referred to by the traffic rate, followed by the prioritize BGP traffic value, followed by the hash BGP traffic value. For example "150NN" would mean the simulation with 150 packets/second, no BGP priority, and no BGP hashing.

As a further point of syntax, when link utilization percentages are being discussed it is usually in the aggregated sense. That is, the utilization figures reported are for the duplex link: X → Y and Y → X. This aggregation will be denoted by a hyphen as in X - Y. However, when discussing factors pertinent to only one-half of the duplex link, the arrow will be used.

Figure 7 gives the global hop count data, the global end-to-end delay value of data traffic, and BGP traffic delay values for nodes 3, 11, and 12 (these three nodes were selected for their involvement in a bottleneck and will give worst case performance data). The averages in Figure 7 are global averages. For the Average Hop Count and Average Network Delay results, the data are averages of averages. The data points for Average Hop Count and Average Network Delay are gathered at evenly distributed times throughout the simulation based on a sliding window as explained in the section below. The data for Average BGP Traffic Delay is based on an average of the age of all BGP packets that were created during the simulation and are first order averages. Refer to Figure 7 as needed during the following discussion of the simulation results.

Average Hop Counts

Traffic Rate/ Configuration	150 Pkt/s	300 Pkt/s	450 Pkt/s	600 Pkt/s
NN	2.0013	2.0054	2.0495	2.0658
NY	1.958	1.9641	2.0061	2.0302
YN	2.0013	2.0054	2.0471	2.0594
YY	1.958	1.9641	2.0085	2.0201

Average Network Delay

Traffic Rate/ Configuration	150 Pkt/s	300 Pkt/s	450 Pkt/s	600 Pkt/s
NN	0.2527	0.2653	0.5307	2.5722
NY	0.4776	0.4911	0.7873	4.073
YN	0.2525	0.2654	0.5507	2.6999
YY	0.4777	0.4912	0.6883	3.414

Average BGP Traffic Delay

Traffic Rate/ Configuration	150 Pkt/s	300 Pkt/s	450 Pkt/s	600 Pkt/s
NN				
3	0.0027	0.0058	0.0208	0.5586
11	0.0031	0.0119	0.0367	2.1315
12	0.0084	0.0181	0.3401	1.612
NY				
3	0.0027	0.0061	0.0366	0.7281
11	0.0042	0.0122	0.0939	1.9954
12	0.0078	0.0198	0.3453	1.9268
YN				
3	0.0026	0.0052	0.0098	0.0137
11	0.0031	0.0111	0.0232	0.0322
12	0.0073	0.015	0.0313	0.0411
YY				
3	0.0026	0.0052	0.0101	0.0139
11	0.0042	0.0121	0.0232	0.0321
12	0.0074	0.0167	0.0322	0.0441

Figure 7. Simulation Results by Traffic Rate and Configuration

The 150 Packet/Second Simulations

The 150NN simulation is the baseline for comparison. It correlates to the conditions found in the internet today and is representative of non-loaded, steady-state network operation. The average end-to-end delay was 0.2527 seconds. The delay data for this simulation was normally distributed about the mean with 100% of the data falling within three standard deviations. The small variance of 0.0102 seconds in the data is indicative of a well-behaved system. The delay values for the three other 150 packets/second simulations were also normally distributed with similar variances. The delay data points are an average of global end-to-end delay of all data packets in the network. An average is gathered every two simulation seconds. The averages are based on a 10 second sliding window. The observed hop count value was 2.0013, which is 94% of the analytical hop count value of 2.1212 (see Eq 10). This indicates a high link availability figure. That is the case, as all links were available for 100% of the time. The hop count data is based on 1400 data points. An average of the hop count of packets arriving at any of the 14 nodes in the network is recorded every two simulation seconds. The averages are based on a 10 second sliding window. One hundred samples are gathered at each of the 14 nodes.

Throughout each of the 16 simulations, as traffic rates increase, portions of the network exhibit pathological behavior. This happens where transmission queues fill and begin to reject incoming messages. The traffic is entering the queue faster than it can be emptied. During this state, the queue remains filled to capacity and the link utilization remains at or very near 100%. An indicator that the network is reaching a pathological state is a non-linear increase in the average link utilization rate as the traffic rates increase from 150 packets/second, to 300, 450, and 600. Similarly, within a simulation, an indication that the network is entering a pathological state is a sharp increase to 100% utilization for links in the network. For the 150NN simulation, the link utilization data for all links was non-increasing over the length of the simulation cycle. The links most prone to failure as traffic rates increase in the model are links 2 - 7, 3 - 12, 10 - 11, and 11 - 12. These had an average utilization rate of 20.2%, 22.2%, 12.7%, and 18.6% respectively.

The other dependent variable of interest in the research is BGP traffic delay. This was measured as the difference between the time the packet was delivered to the destination node and the time the packet was created. The measurements were taken at three of the worst performing links in the model to obtain "no worse than" figures, (links 3 -12, 10 - 11, 11 - 12). These links represent a bottleneck topology in the model (see Figure 6), and can be expected to fill most quickly. An increase in the age of the BGP traffic is indicative of queuing delays as the links and queues fill.

In the real internet, where the BGP message would contain pertinent link information, delay in delivery could lead to injection of stale topological data, produce transient routing loops, and eventually lead to flapping storms as the delays increase and sessions time out. For the 150NN simulation, no increase in BGP message age was observed over the life of the simulation. The average age for all BGP messages arriving at node 11 was 0.0031 seconds, 0.0084 seconds for node 12, and 0.0027 seconds for node 3. These values are reasonable given the small packet size of 152 bits for the predominant BGPK message and the small average delay-through-queue values on the links for the 150 packet/second simulation.

As an example, the expected BGP message size, based on the BGPK/I message distribution explained in Chapter 4, is [(152 * .91) + (1120 * .09)] = 240 bits. The transmission delay of this packet from node 11 to 12 is 0.0026 seconds, the average delay through the transmission queue for all packets on the 11 → 12 link (which is a data item that was available from the simulation), was 0.0024 seconds, and the propagation delay from node 11 to 12 is 0.000118 seconds. The totals of the values from node 11 to 12 is 0.005118 seconds. The corresponding value from node 3 to 12 is 0.007143 seconds. So the values of BGP packet ages appear to be reasonable in the model.

Figure 8 plots the delay of data packets in the network over time. The network end-to-end delay result for the 150NN and 150YN simulations are nearly the same, so only the chart for the 150NN simulation is presented.

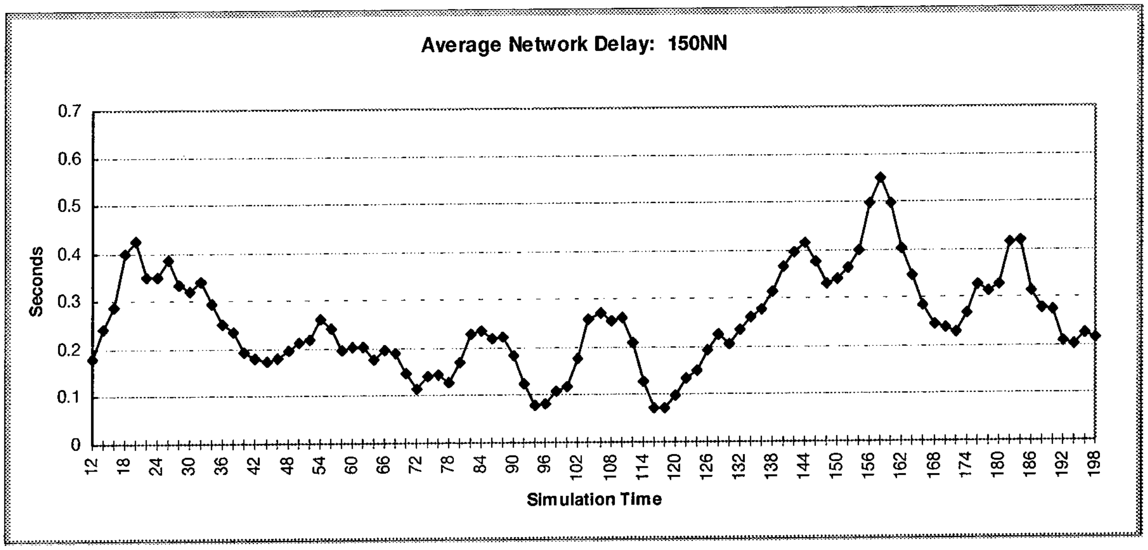


Figure 8. Average Network Delay for the 150NN Simulation

The 150NY simulation was similar in all respects to the baseline simulation with the obvious exception of end-to-end delay. The delay value was 0.4776 for a 189% increase over the baseline. This value has two contributing factors that strongly suggest it is a worst case scenario (reference the Node Out Degree subsection of the Instantiation of Particular Model Parameters section in Chapter 4 for an explanation of the delay associated with the use of hashing in the Transit-Stub ISM). First, the packet delay that data traffic can expect to incur while crossing a router that is currently performing a hash of a BGP message is intentionally set high; secondly, the amount of BGP traffic in the model is also intentionally set high.

The average hop count value for the 150NY simulation was 1.958. This value is within 3% of the hop count value for the 150NN simulation. Again, no BGP peering sessions were dropped and link availability was 100%. The link utilization values were similar, exhibiting less than 1% variation in all cases. As in the 150NN simulation, the network remained in steady state until it terminated. There were no queue overflows, nor were there any indications of a pending pathological state. The increase in end-to-end delay can be solely attributed to the cost of hashing BGP messages. The average age of BGP messages arriving at nodes 3, 11, and 12 was

0.0027, 0.0042, and 0.0078 seconds respectively. This compares favorably with the values of 0.0027, 0.0031, and 0.0084 seconds of the 150NN simulation.

The 150YN simulation data is unremarkable. The impetus for prioritizing BGP traffic is to ensure steady and reliable network topology information as traffic rates increase. The nominal traffic rate of 150 packets/second makes prioritizing BGP traffic unimportant. The network delay was 0.2725 seconds compared to the 0.2527 seconds of the 150NN simulation. The other parameters were also nearly identical to the baseline simulation. The hop count value was 2.0013 which is identical to the hop count value for the 150NN simulation. Link utilization values had less than 0.001% variation. The BGP traffic delay values for nodes 3, 11, and 12 were 0.0026, 0.0031, and 0.0073 seconds respectively. These values are equal to or less than the values of the other simulations where BGP traffic wasn't prioritized. The difference is small but nonetheless expected because BGP traffic is moved to the front of the transmission queues. The small difference is due to the small delay-through-queue values in the 150 packet/second simulations. The 150YN simulation maximum queue occupancy at the worst link was 7 packets. The maximum packet occupancy of the queues on the largest links was 1. This produced average queue delays that were no worse than 0.0047 seconds.

The 150YY simulation had a network average end-to-end delay value of 0.4777 which is nearly identical to the 0.4776 value observed in the 150NY simulation. The link utilization rates were varied less than 1% from the baseline. The hop count value was 1.958 which is identical to the 150NY simulation and within 3% of the baseline simulation. The BGP delay for nodes 3, 11, and 12 was 0.0026, 0.0042, and 0.0074 respectively. Figure 9 presents the data packet delay values for the 150YY simulation. Note that the data packet delay values are very similar in the 150NY simulation, so only one chart is presented. Also note that the shape of the plot closely resembles the that in Figure 8, just shifted in magnitude.

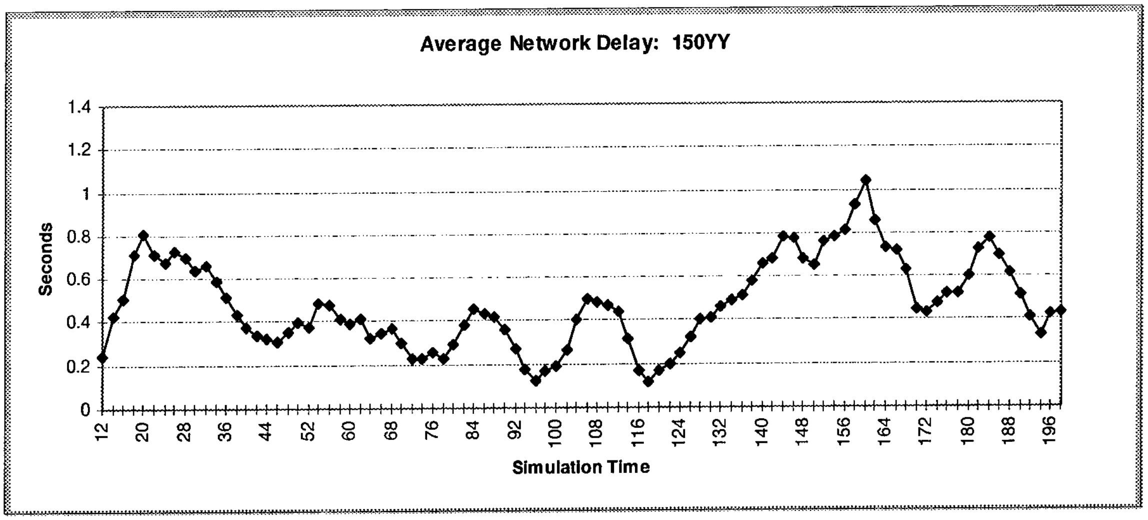


Figure 9. Average Network Delay for the 150YY Simulation

The 300 Packet/Second Simulations

The 300 packet/second simulations did not substantially change any of the dependent variables of interest. The network behavior remained in steady-state for the entire time, which is to say, none of the observed factors that are indicative of pathological network behavior, were steadily increasing over time. No transmission queues became full. The link utilization rates were very nearly twice the utilization rates for the 150 packet/second simulations (note the linear relationship). The network delay values increased by an average of only 3.9%. The increase in network delay is attributable to the increase in average delay in the transmission queues, which for the most congested links increased by factor of 2 to 3 but did not increase at all for the larger links.

The average age of BGP messages delivered to nodes 3, 11, and 12 increased by factor of 2.401 between the simulations that prioritized the BGP messages (150YN / 150YY → 300YN / 300YY). Since the BGP messages are only one-hop messages in the network, the percent of increase in delivery time is greater than the corresponding percentage increase in network average end-to-end delay. Put another way, the BGP delay is so small compared to the delay of the data packet, that an increase in average queue delay will yield a larger percentage increase in

BGP traffic delay than in the data traffic delay. Since the network remained in steady state and exhibited no pathological behavior due to loading, the age of BGP traffic that was not prioritized was only 12% greater than the BGP traffic that was prioritized. At the 300 packet/second level of loading, there is still no compelling reason to prioritize BGP traffic. Figure 10 gives the data packet delay data for the 300NN simulation. It is remarkably similar to the 150NN plot, just shifted slightly in magnitude. Note that the similarities in the delay curves for the 150 and 300 packet/second simulations are directly attributable to the fact that the same global random number seed was used. Changing the global seed will yield differently shaped curves.

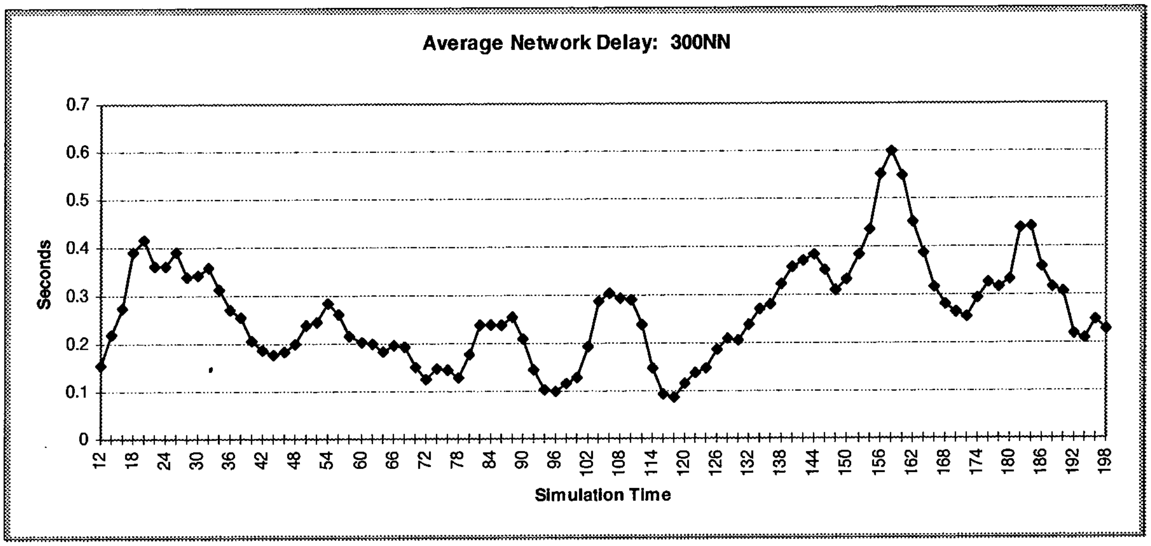


Figure 10. Average Network Delay for the 300NN Simulation

The 450 Packet/Second Simulations

This level of loading began to produce pathological network behavior. Link 3 - 12 did not saturate, but saw an increasing utilization percentage over the life of the simulation. Link 12 → 11 began to saturate at the 112 second simulation point, and reached saturation at approximately the 120 second point (see Figure 12). The rest of the network links operated in steady-state. No BGP sessions timed out because of non delivered packets. It is not clear at this point if a prolonged simulation would cause BGP session oscillation, but initial conditions are favorable for this condition to occur.

It becomes apparent at the 450 packet/second loading that the BGP traffic situation is helped by the fact that peering sessions are established with one-hop neighbors. The non-prioritized BGP traffic delay increases on the links to nodes 3, 11, and 12. But even though the average end-to-end delay increases to 0.6393 averaged across all four simulation configurations at the 450 packet/second rate, the non-prioritized BGP traffic delay stays well within the packet timeout values for all links with the exception of those connecting to node 12. The BGP traffic delay value to node 12 is 0.3427 seconds which is well above the timeout values for link 3 → 12, or 11 → 12. Considering that the overall BGP delay is increasing, it appears that BGP session oscillation will occur at some later time (outside of the 200 second simulation window).

The 450NN simulation produced an overall network delay value of 0.5307. The data is not normally distributed because the network does not achieve steady-state operation and the delay data is increasing over the life of the simulation. This is the case for the 450NY/YN/YY simulations also. Figure 11 shows the average data packet delay data for the simulation. Like the 150 and 300 packet/second simulations, the shape of the delay plots between the 450 packet/second simulations are very similar.

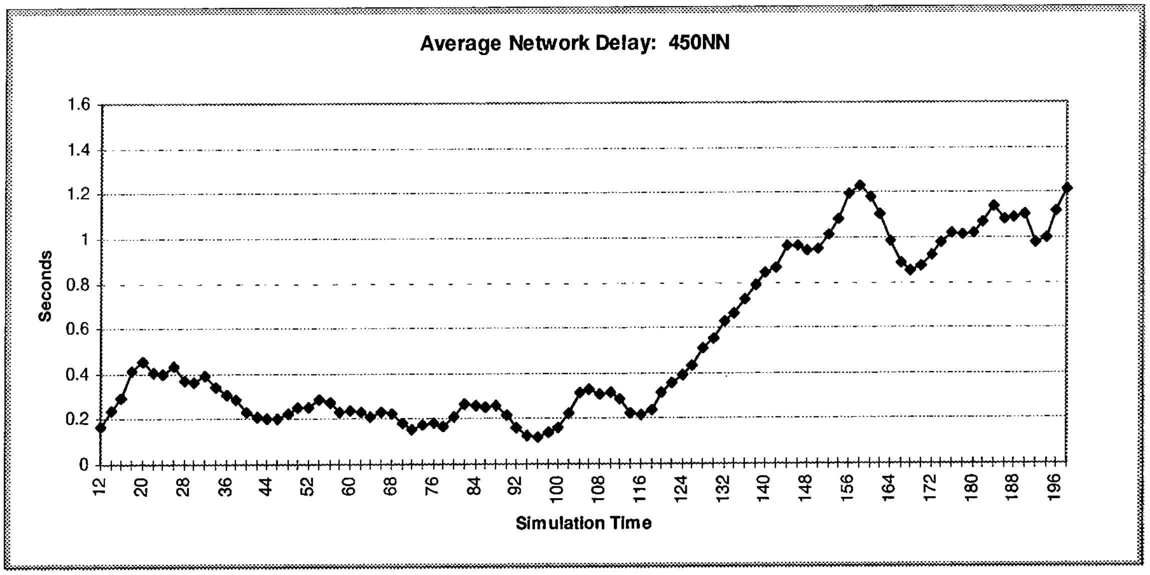


Figure 11. Average Network Delay for the 450NN Simulation

The hop count value for the 450NN simulation was 2.0495. This is 2.4% larger than the value for the baseline simulation of 150NN but still indicative of link availability and non-oscillation of BGP sessions. Link 12 → 11 transmission queue did not fill and begin to reject packets, but the average delay through the queue of 1.1052 seconds caused successive requests for retransmissions on the link thus increasing the link utilization. The maximum number in the 12 → 11 transmission queue during the simulation was 598 packets. No other link experienced saturation, but the utilization rate for link 3 → 12 began to steadily increase at the 112 second simulation point. The rest of the links had non-pathological utilization rates that were a 3x extension of the utilization rates in the 150NN simulation. That is, the linear relationship of utilization rates between simulations of increasing traffic rates stayed intact which indicates steady state operation. By contrast, the link 3 - 12 utilization percentages for the 150NN, 300NN, and 450NN simulations were 22.24%, 44.38%, and 74.56% respectively. The link 11 - 12 utilization percentages were 18.63%, 37.81%, and 68.29%. Figure 12 shows the link 11 - 12 utilization rates. The non-linear growth in link utilization rates is indicative of pathological behavior.

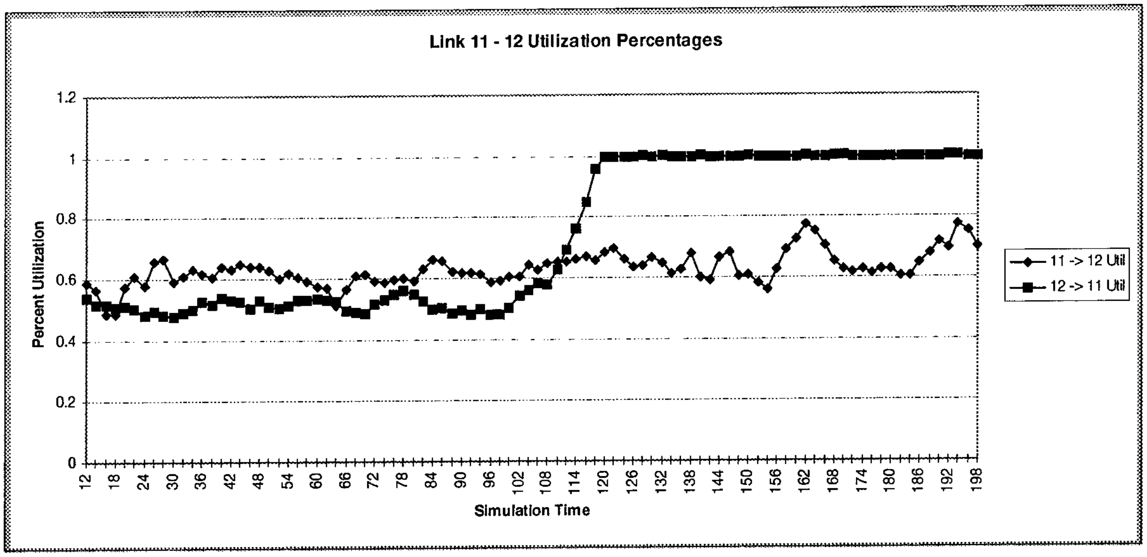


Figure 12. Example of Link Saturation in the 450NN Simulation

The 450YN simulation results were very similar to the 450NN simulation. The average hop count was 2.0471 which was within 0.2% of the 450NN simulation. Following the trend established in the 150, and 300 packet/second simulations, the network delay plots for all four simulations were very similar in shape. The average network delay for the 450YN simulation was 0.5507 seconds which is within 4% of the delay value in the 450NN simulation. The BGP traffic ages for this and the 450YY simulations remained low because prioritizing the traffic prevented long delays in the transmission queues. Unlike the 300 packet/second simulations, the benefits of prioritizing BGP traffic become apparent at the 450 packet/second traffic rate.

The largest surprise in the data occurs in the network delay value for the 450YY simulation. Following the trends for the 150 and 300 packet/second simulations (see Figure 7), the 0.6883 delay value would be expected to be close to the 0.7873 value found in the 450NY simulation. The reason that it is 12.6% lower is because at data rates of 450 packets/second and higher, the network begins to link timeouts occur causing retransmission of packets. This includes retransmissions of BGP traffic in simulations where that traffic is not prioritized. Because the Transit-Stub ISM has a sparse transport layer representation, there is no ability to relate a retransmitted packet to its original and both are eventually processed. This means that BGP messages are re-processed in the network layer of the model producing an artificially raised level of network delay. Where priority is applied to BGP traffic, no retransmissions occur because no BGP packet times out at the data link layer. This leads to a more accurate reduced overall network delay. The network delay figures for the 450/600 YY simulations are more accurate than those of the 450/600 NY simulations.

The 600 Packet/Second Simulations

These simulations exaggerate the behavior of the 450 packet/second simulations. The differences are in the level of pathological network behavior exhibited, and the time within the simulation that the pathological behavior is first manifest. The observance of pathological behavior as evidenced by the link utilization factors occurs at the same time within the 600NN, 600NY, 600YN, and 600YY simulations. The lone difference in the measurement of dependent variables, is the average utilization of link 2 - 7. In the simulations where hashing was used, the utilization rate was 93%. It was 84% in the simulations where hashing was not used. In the collection of data from the simulations, there is no evident causal relationship that can be proposed for this difference. However, in both cases, the utilization rates on link 2 - 7 were steadily increasing during the simulation and would become saturated sometime outside of the simulation window.

The network delay data in the 600 packet/second simulations was steadily increasing from the first observed instance of pathological behavior. Figure 13 shows the shape of overall network delay of data packets.

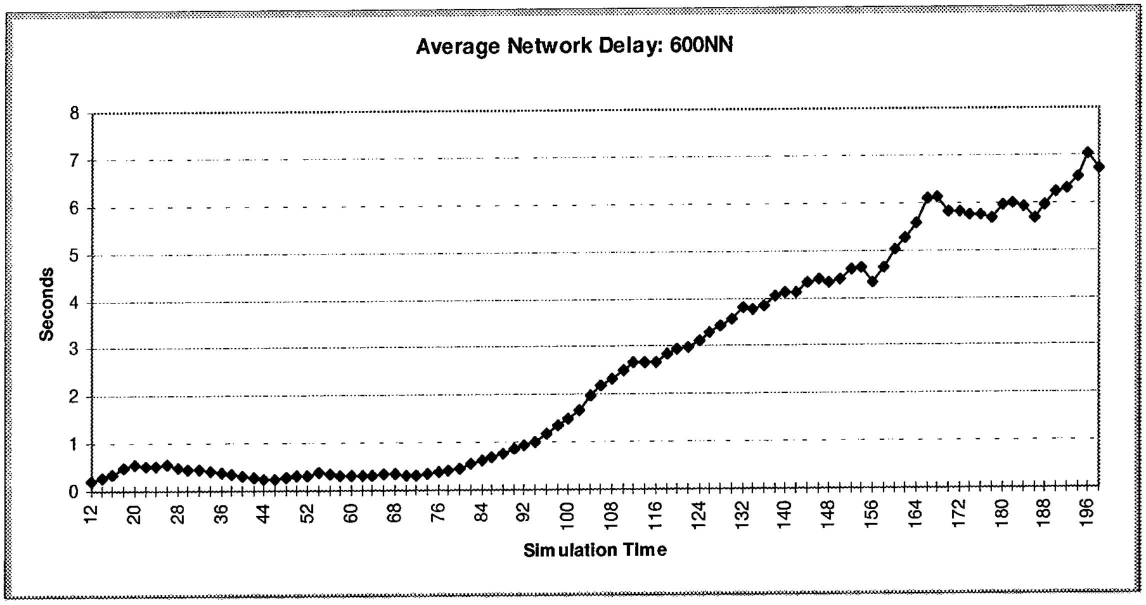


Figure 13. Average Network Delay for the 600NN Simulation

The pathological network behavior begins at about the 84 second simulation point. This corresponds to the beginning of link saturation in the 2 - 7, 3 - 12, and 11 - 12 links. As an example, the link saturation plot for link 11 - 12 is shown in Figure 14. Contrast this with Figure 12 to see the effects of the extra loading produced by the 600 packet/second simulation.

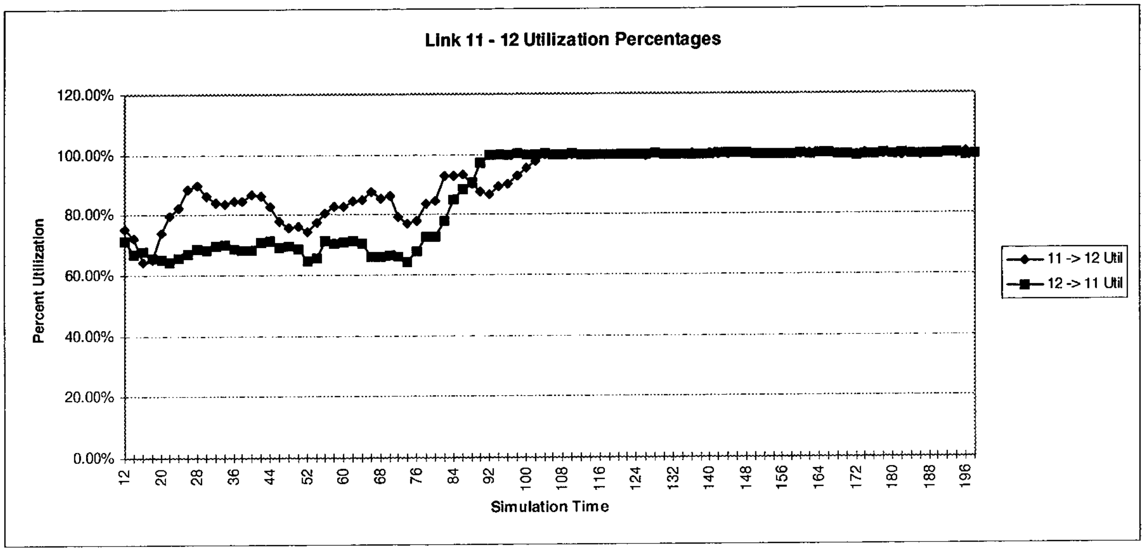


Figure 14. Example of Link Saturation in the 600NN Simulation

The largest difference between the 450 and 600 packet/second simulations is in the average delay of BGP traffic. The delay value for BGP packets increases over time during the 600 packet/second simulations for all three critical nodes of 3, 11, and 12. All three critical nodes had BGP traffic delay greater than the link timeout value in the 600 packet/second simulations where priority was not applied. The BGP traffic delay value for the critical nodes did not cross that threshold where priority was applied. By contrast, the only node at which the BGP data traffic delay increased beyond the link timeout value for the 450 packet/second simulation is node 12 whether or not priority was used.

The plot in Figure 15 was created by capturing all BGP traffic in the simulation and subtracting the time the packet was created from the current simulation time and displaying the difference. Note that the plot is missing six data samples of BGP traffic for the 600 packet/second simulation and one for the 450 packet/second simulation. In the 150 and 300 packet/second simulations, 80 BGP messages were delivered to node 12. Because the same global random number seed was used between all the simulations, the 450 and 600 packet/second simulations also had 80 BGP messages generated from all peering nodes of node 12. However, in the 450 packet/second simulation, one data sample is missing, and in the 600 packet/second simulation, 6 data samples are missing. The non-delivered BGP messages are due to: 1. the delay of BGP messages where that delay pushed the delivery of the BGP message outside the simulation termination time of 200 seconds (as is the case in the 450 packet/second simulation verified from the fact that there were no dropped packets on the link 11 → 12, but still one missing message as evidenced by the last triangle in the plot in Figure 15 corresponding to the 79th Sample Number), and 2. a mixture of the reason put forth in (1.) above, and the fact that the transmission queue on link 11 → 12 began rejecting messages as it became saturated. The last "x" in the plot corresponds to the 74th Sample Number for the 600 packet/second simulation.

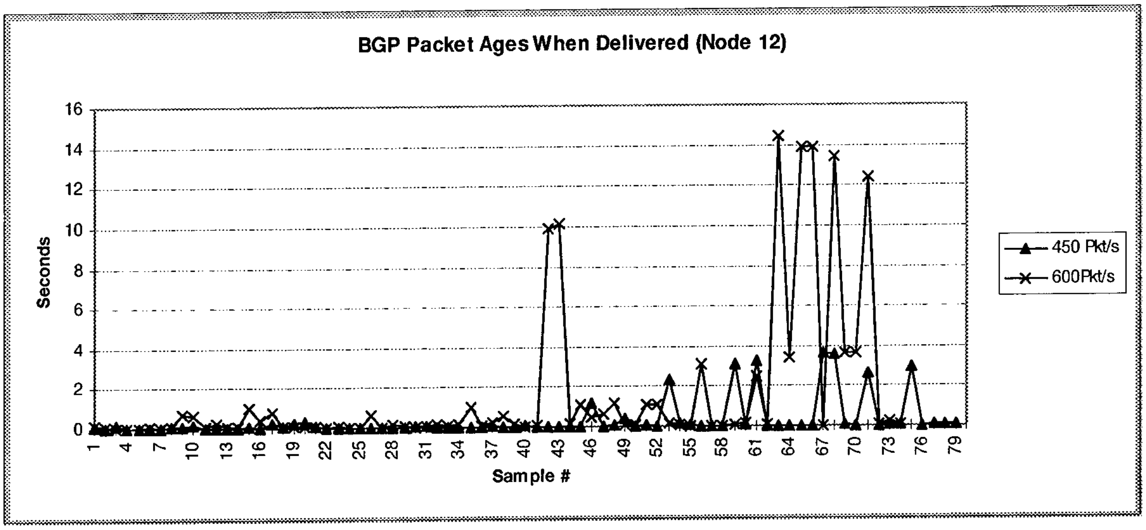


Figure 15. Increase in BGP Traffic Delay for the 450 and 600NN Simulations

In the 600 packet/second simulations it is clear the network is in a pathological operating state and will remain there. The increase in BGP traffic delays to nodes 3 and 11 are similar to those at node 12. Note that the shape of the BGP traffic delays for the 600 Pkt/s line are from all peering nodes to node 12. That is why there are dips in the shape. The highest delays are all coming from node 11 and the secondary delays from node 3. Although no BGP session timeouts occurred in the simulation because the 200 second limit did not allow the 90 second BGP timer to expire, it is evident from the data that the environment is amenable for BGP session oscillation to occur. The only transmission queues to fill and reject packets in the simulation were the ones on the 11 - 12 link. The 11 → 12 link rejected a total of 2087 packets and had an average delay through queue value of 2.68 seconds. The 12 → 11 transmission queue rejected 13217 packets and had an average delay through queue value of 4.33 seconds. The 11 → 10 transmission queue had an average delay through queue value greater than the average link timeout period in the model (0.44 compared to 0.1375 seconds), and the queues on the 3 - 12 links both had average delay through queue values exceeding the average link timeout period.

The benefits of the use of priority for BGP traffic are immediately evident within the 600 packet/second simulation when observing the BGP traffic delay at the critical nodes of 3, 11, and

12 (see Figure 7). In the 600YY simulation, data traffic end-to-end delay averaged 3.414 seconds. The average delay of BGP traffic node 12 was 0.0441 seconds demonstrating that even during the busiest times, BGP information will be reliably delivered. The data for the 450YY simulation is shown for comparison in the following figure.

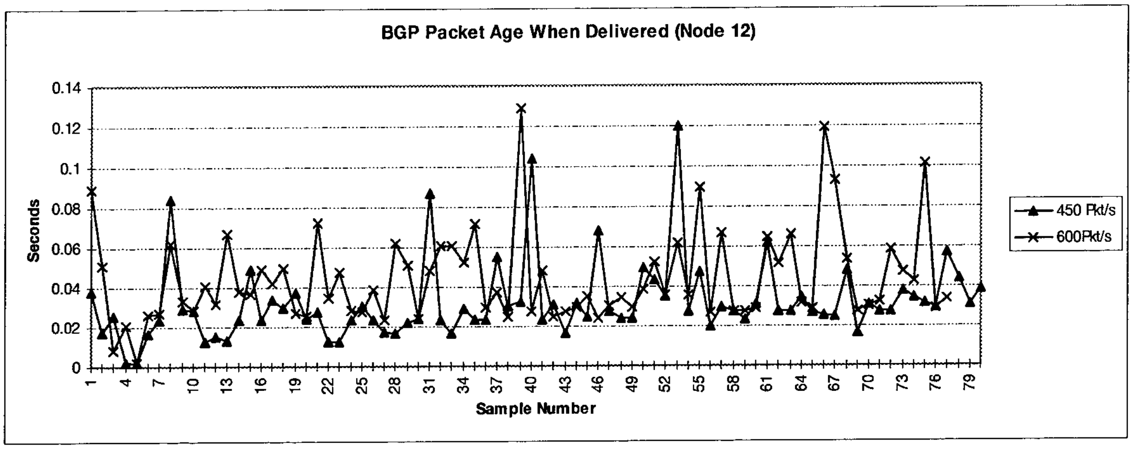


Figure 16. The BGP Traffic Delay Picture for the 450 and 600YY Simulations

Discussion of Overall Results and Model Artifacts

The data for the comparing the outcomes of all simulations is summarized in Figure 7. The hop count data remained well behaved throughout the simulations. The relatively constant numbers were indicative of overall link steadiness. Even though the 600 packet/second simulations produced pathological network behavior, there were no BGP sessions lost and therefore no network recomputations took place in the model that could have altered the hop count appreciably. Two trends emerge from looking at the hop count data: the hop counts for the simulations where hashing was used were smaller than for the simulations where no hashing was used, and the hops counts increase between the 150 packet/seconds simulations to the 600 packet/second simulations anywhere from 2.9% to 3.7% depending on the configuration. Given that the variance in the hop count data was 4%, the increase in hop count due to traffic rate does not merit concern. Although untested, a possible explanation for the difference is that the shape of the population over which the 10-second windowed averages were taken changed as the traffic rates increased. This circumstance could have also affected the hop count data when hashing was used because hashing alters the delay of data traffic on the network.

The network delay data is not as reliable for the 600NN and 600NY simulations as for the 600YN and 600YY simulations. This is a model artifact due to the minimal transport layer representation and the abstraction of actual packet data into a Packet Length field in the packets. The model does not know how to tell the difference between a retransmitted packet and the original. The one-deep retransmission capability in the model was introduced to model the extra traffic that is injected onto the links as timeouts occur and retransmissions requested. These timeouts occur because of excessive queuing delays as the network becomes busier. The model actually processes both the retransmitted packets and the original packets if one or the other are not rejected entirely from the transmission queue because it is full. This leads to artificially high processing delays associated with receiving extra BGP messages. This increases the global delay as data packets that traverse the BGP nodes now have a higher likelihood of being delayed while that node is processing a BGP message. This inaccurate delay representation is alleviated in the simulations where the BGP messages are processed with priority as requests for retransmission of the BGP messages do not occur.

The average BGP delay data in Figure 7 shows a general growth trend as the data rates increase. This is an expected trend regardless of whether priority is assigned to BGP messages or not. This is because the likelihood of waiting in the queue increases as the traffic rates increase when considering the simulations where no priority is assigned BGP traffic. In the simulations where priority is assigned, the likelihood of finding a packet currently in transmission as a BGP message enters the transmission queue increases with the increase in data traffic.

Conclusion

In considering the data, a firm case can be made to introduce priority for BGP traffic. A further consideration should be made to not only prioritize BGP traffic, but to also allow it to preempt any packet of lower priority currently being transmitted at the precise moment the BGP packet arrives. The data for the 600YY simulation shows that even though BGP traffic is prioritized, 3 packets out of 80 destined for node 12 were rejected at the data link layer because the arriving BGP packet found the queue full. This can be observed by looking closely at Figure 16. Note that the last "x" corresponds to the 77th Sample Number. Also observe that the data for Average BGP Traffic Delay in Figure 7 increases with traffic load despite the fact that priority is assigned to the traffic. This is due to the increasing likelihood that an arriving BGP packet will find that the transmission queue is currently transmitting a packet or full. In either case, a priority with preemption scheme would disallow dropped BGP packets and decrease the delay even further. The use of priority for BGP traffic illuminates the research problem as stated in level one of Figure 3. The use of priority will help stabilize the operation of the BGP protocol during busy times on the internet thereby alleviating the natural DoS conditions that occur because of increasing traffic rates.

To address the second part of the research problem (see level 2 of Figure 3), the costs and benefits of hashing have to be explored. The cost is shown by using the average of the delay data for the 150/300/450YN simulations and comparing that to the average of the delay data for the 150/300/450YY simulations. There is an increase of 155% in the network average end-to-end delay due to hashing BGP traffic. Factoring in the 600 packet/second simulations yield a 135% increase. But because the 600 packet/second simulations represent a steady pathological state of operation, it would not be accurate to include those outcomes in the comparison of delay times as the internet exhibits regional pathological behavior but no global instance of pathological behavior has been observed.

It is unclear if the increase in cost incurred as a result of hashing BGP traffic will outweigh the benefit of protecting the infrastructure from DoS attacks. Even as a worst case scenario, a 155% increase in delay may seem a prohibitive cost. This cost is not set in stone however. There are alternatives that may be used to mitigate it. One is to hash only the BGP protocol on the backbone portions of the internet where equipment from the Routing Arbiter [13] project is already in place. This would ensure that at least the global internet information maintained by core backbone routers could be protected against DoS attacks instigated at that level. It does not ensure that BGP information received from the lower level peers can be similarly trusted however. And considering the trends illustrated by Table 3, where connections to the upper hierarchical layers in the internet are becoming more prevalent from the lower hierarchical layers, this strategy may not be as effective as it could be otherwise.

A second alternative may be to only hash the BGP update messages themselves and accept the risk of a spoofed BGP keepalive message. This could be a promising strategy because the processing of BGP keepalive messages is minimal as they are only 152 bits in length and serve only to reset the session timer. Actually, in the busiest traffic times, bogus "spoofed" keepalive messages, instead of having the effect of increasing the processing burden of the target machine, may be more of a benefit than hindrance as session timers that could otherwise expire are now kept alive. BGP keepalive messages contain no actual link state information and cannot be used in a DoS attack to inject bad routing information into the internet. In addition, the BGP update traffic represents less than 10% of the total BGP traffic when the protocol is operating normally. So introducing hashing of the update portions of the traffic would mean a reduction of the burden associated with hashing BGP traffic on the internet and would have associated reductions in average end-to-end delay. A simulation run that tested this scenario yielded a 132% increase in average network end-to-end delay.

VI. Conclusion

Introduction

This chapter will present concluding remarks about the model construction and simulations/analyses. It will also list some suggestions for model improvements that are designed to more fully represent the problem domain. It is intended for future master's degree candidates and others who may want to do follow-up research. To than end, Appendix B contains the history of model modifications as it was being built. The format of the modification history is not formalized. It is a copy of my personal modification log and is intended to show the standard pitfalls one can encounter while building a model of this complexity and to act as an aid to anyone wishing to continue to refine the model, especially if it will be done using BoNES Designer.

Alternate considerations that bear on the research problem will also be introduced and discussed briefly. The purpose is to round out the approach by introducing topics that are important but that do not necessarily fit in with the engineering thrusts of this work. Finally summary remarks are given.

Conclusions

The data presented in chapter five is preliminary. At this point, it can only be used in a suggestive nature. Prioritizing BGP traffic clearly counteracts the environment in the internet that is amenable to flapping storms. The cost to implement priority is cheap, but it has to be done globally to have the desired effect. The question of hashing BGP traffic is more complicated. It is expensive in terms of packet delay, there is no industry consensus on implementation techniques, and the DoS threat is not yet quantified.

The main thrust of this research was to build a representative model. The construction of the model and parameter instantiation were judged to be representative of the problem domain [3]. Also adding to the validity of the model are direct inputs from experts in the field. The model parameter instantiation relied heavily on such inputs. Also, the BGP packet age results, though based on preliminary simulation runs, were observed to be close to results obtained by more elaborate vendor-level research, as commented by Ed Cain during the defense.

Follow-up Research

The analysis in Chapter 5 was based on too few runs of the simulation to obtain confidence in the data. A good first step in follow up research is to obtain more data from the model so confidence can be established. The results are preliminary and are meant to establish an overall framework for future research in this area.

Other extensions to the model include building transport layer entities for it. This would include the capability to emulate the TCP protocol with such functions as dynamic adjustment of retransmission window sizes and timeout values for ACK packets on the links based on current traffic flow. This would entail introducing specific data and acknowledgment packet numbers and keeping track of them during the simulation. Adding this capability would make the model more representative of the internet domain and subsequent results more germane to the problem areas.

Another extension to the model would be to run some simulations of 21K second lengths at the 450 packet/second rate to try and force a routing storm situation. This would emulate the time period of high traffic rates observed on the North American section of the internet where traffic rates stay very high between 16:00 and 22:00 hours. It would be educational to gather information about the behavior of BGP traffic during a routing storm running simulations with and without BGP priority. A rough estimate of the amount of time that this simulation would take is 27 days using the BoNES Designer program executing on a Sun Sparc Station 20. Before a simulation of that length is attempted, it would be wise to adjust the probes in the simulation so they trapped less data. Otherwise file I/O errors will occur as the program output exceeds allotted storage sizes.

Alternate Considerations

The use of priority and hashing for BGP traffic is not a new idea. The specification for the Border Gateway Protocol 4, RFC 1771 [22], recommends using priority for the protocol. The Marker field in the header of BGP messages is a container that can hold the hash of a message. It was designed with that purpose in mind. The reality though, is that neither is used in practice. Whether or not these mechanisms are employed for the protocol, there are some practical considerations that network administrators can take to ensure the efficacy of their network and others. Some of these are:

- Run a standard intra AS protocol so that a consistent picture can be maintained and the interfaces to BGP can be maintained consistently. This will reduce the chance of passing unintended or harmful intra-AS path information into the inter-AS domain. Simpler is better.
- Do not mix BGP protocol versions. The general consensus is that BGP3 is no longer in use. But the Air Force InterNetwork (AFIN) interface to the internet from the Wright Patterson AFB AS is running BGP3. This is wasteful of processing as the BGP3 protocol does not support IP address aggregation. The interface between BGP3 and BGP4 will introduce overhead between the peers as all address information has to be converted from the specific 4-octet full representation to an aggregated form or vice versa depending on the traffic flow. Depending on the peering policy (exactly what BGP information is being shared between the routers), this has the potential to introduce

unneeded processing overhead on the routers. This could also lead to errors because of the extra overhead in building and maintaining the routing tables on the AFIN router.

- Secure Trivial FTP (TFTP) Servers. Many network administrators hold a picture of their router's routing tables on a TFTP server for ease of reloading in case of a system crash. The TFTP program has no security measures and it will grant any user-id access without demanding a password. All a hacker need have is the address of the server to gain access to sensitive routing information. Even if access lists were used to keep out traffic to the server from untrusted domains, a spoofed session will still pass the access lists and access will be granted. At the very least, an administrator should move the backup copies of routing tables off TFTP servers. The draconian measure is to disallow all router access via FTP or Telnet and only allow access through the console port. Accessing the router through the console port requires physical access to the router. The TFTP server would be the first place that a hacker seeking to instigate a DoS attack would probe
- Install access list filters in the routing tables that disallow any packets originating from within the intra-AS domain and having an IP address different from those belonging to the domain to exit that domain. This will absolutely disallow IP spoofing attacks from within the host domain. The Wright Patterson AS is fully protected in this sense.
- By all means, switch to SNMP version 2. The older version of the protocol does not encrypt passwords that are needed to gain access to the routers. A network sniffer set to filter SNMP traffic could obtain these passwords in short order. One particularly sensitive attribute that is accessible through an SNMP session is the max number of hops field allowed in the IP packet before the router throws it in the bit bucket. If an hacker obtained the SNMP session password, he or she could effectively kill all traffic exiting the router by setting the value for that field to 1.

These are a few mostly painless procedures that should implemented by a network administrator to heighten the security posture of the internet. Even if such ideas a prioritizing and/or hashing the control traffic of the internet never gets any further consideration because increasing commercialism and heterogeneity prohibit a affective global administrative scheme, measures such as those outlined above should be aggressively sought out and implemented in networks that are under DoD control.

Summary Remarks

The old moat style security paradigm has to be challenged and re-thought if we expect to maintain Information Superiority into the twenty first century. As the DoD embraces the natural robustness that is found in the free market ... certainly a time-proven formula ... so it must accept the risks of less control over the medium that it shares with its civilian counterparts.

The benefits gained by relying on COTS solutions have the risk of tying the hands of those whose job it is to ensure security and survivability when the clear solution is clearly outside their span of control. If we are to rely on the commercial internet to carry the daily business data of the DoD, then we have to be very clear in our understanding of the risks involved. The DoD is certainly in no position to mandate that the infrastructure be made unassailable to DoS attacks by insisting that the control traffic be prioritized and protected. And as time passes and the DoD continues to downsize, our aggregate purchasing power and market punch will continue to diminish. We may be doing a good job of maintaining the moat, but our adversaries may have the ability to starve us.

Appendix A: The Transit-Stub Internet Survivability Model

The Transit-Stub ISM is presented here by layers. These layers roughly correspond to the OSI 7-layer communications model. However some layers, like the Init Network, Traffic Generation, and Node-level layers are specific to the model construction. Not every primitive module is shown here. For instance, the modules that read and write the various matrices used in the model are not shown. But any module that has a bearing on model operation is explained here. Furthermore, any module, that has an embedded module where the relationship between the two modules is ambiguous or non-trivial, will be shown. The following table is a road map for appendix A. It contains the list, by category, of all the modules in this appendix.

Table 13. Summary of Modules By Layer

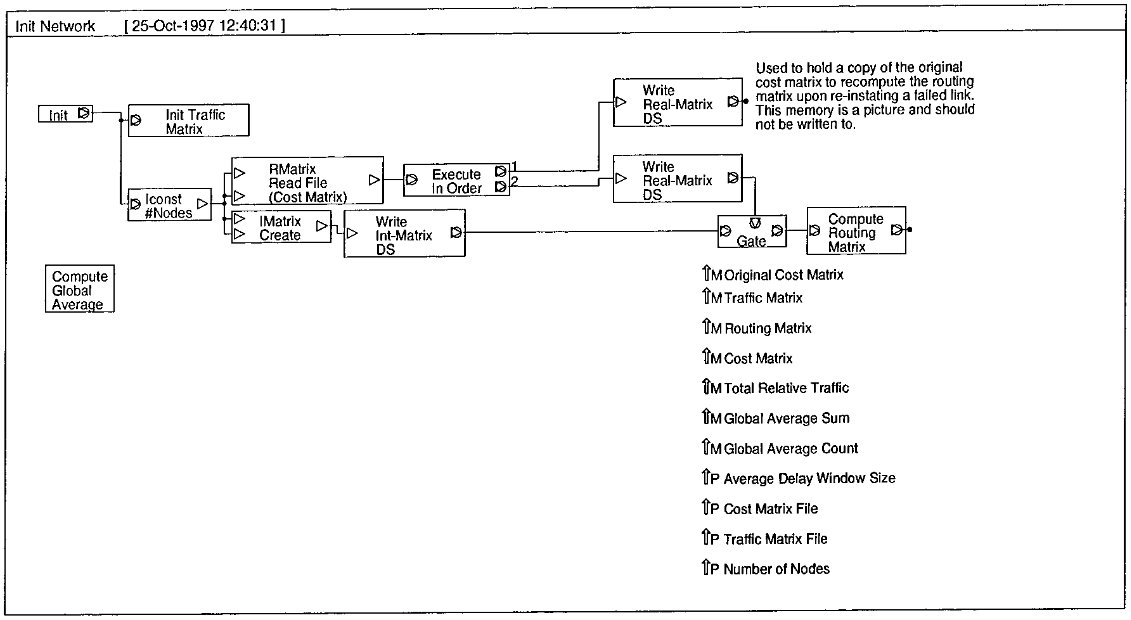
Layer	**Module Name**
Initialize Network	Initialize Network Compute Global Average
Traffic Generation	WAN BGP Traffic Gen Start BGP Traffic Compute BGP Traffic Generate BGP WAN Packet
Physical	Full Duplex Link
Data Link Layer	Data Link Layer Cancel Packet Timer CSU/DSU Behavior Hold Buffer Timestamp Packet Priority
Network Layer	WAN BGP Network Layer WAN Network Layer BGP PDU BGP PDU (Out) BGP Out Processor BGP PDU (In) BGP Memory Test BGP In Processor Fixed Processing Delay (BGPI) Reconfigure Network (Link Up) BGP Timer Expire

Layer	Module Name
	Processing BGPI? Recomputing Network? Reconfigure Network (Link Failed)
Node Layer	BGP/WAN Node WAN Node
Simulation Layer	Transit-Stub ISM Simulation System (14 Node)

Each figure will have an explanation of its function, a description of the data flow, and the parameters of the module will be explained. All parameters are underlined when referenced in the text.

A brief note about terminology: the term "module" and "block" are used somewhat synonymously in this appendix. The difference is that *module* will be used to refer to a logical grouping of designer functions that are non-primitives, e.g., would have several operations to perform on a packet traversing that module. A *block* is a primitive operation on a data structure such as a delay function. In the figure below, a *module* would be Compute Routing Matrix task, while a *block* would be the Gate or Execute In Order functions. When used, the terms *block* and *module* should be clear from the context.

In many cases modules use the same parameters as other modules. This is because some have to perform the same primitive operations on the data structure and also because the parameters are inherited by all lower layer modules that need to access them. So while the same parameters may appear in several modules, they will normally only be explained when first referenced unless the parameter is used in a different way by subsequent modules. Finally, there is an in-depth explanation of certain model parameters in the "Instantiation of Particular Model Parameters" section of chapter 4. Refer to it as necessary.



Used to hold a copy of the original cost matrix to recompute the routing matrix upon re-instating a failed link. This memory is a picture and should not be written to.

Figure 17. Initialize Network

The Init Network module is the first to execute within the simulation. It takes no simulation clock time to complete, but performs preliminary house keeping functions for the simulation. It loads the link cost and traffic matrix memories from separate files that are provided by the user. The routing matrix memory is computed here from the cost matrix memory using the Dijkstra algorithm. During the simulation of the model, if links fail or are re-initialized, the routing matrix memory is recomputed on the fly. The Init Network module also computes the sum of the traffic matrix which is a memory argument used in the Traffic Generation module.

Parameters:

- Original Cost Matrix (memory, non-local): used by the Compute Network (Link Up) module to re-establish the network after a link has failed because of a dropped BGP peering session. An entry in the Original Cost Matrix OCM_{ij} indicates the cost of the link from node i to node j. If there is no connection between the two nodes then the link cost is set to 1E6 which stands for "infinite cost" in this model.

- Traffic Matrix (memory, non-local): Used by the various traffic generation modules (in conjunction with other model parameters) to compute the relative amount of traffic between any two nodes in the network. See the Compute BGP Traffic module for an explanation of its use.
- Routing Matrix (memory, non-local): The Routing Matrix memory is referenced at each node's network layer to make routing decisions for packets traversing that node. This memory is updated by during the simulation of the model is links fail or are re-instated after failing. An entry in the Routing Matrix RM_{ij} indicates the next hop of a packet which is at node i and is destined for node j.
- Cost Matrix (memory, non-local): Updated by the Recompute Network (Link Failed) module and subsequently used to compute a new routing matrix. An entry in the Cost Matrix CM_{ij} indicates the cost of the link from node i to node j. If there is no connection between the two nodes, then the link cost is set to 1E6 which stands for "infinite cost" in this model.
- Total Relative Traffic (memory, non-local): Contains the sum of all the entries in the Traffic Matrix memory. Used by the various traffic generation modules (in conjunction with other model parameters) to compute the relative amount of traffic between any two nodes in the network. See the Compute BGP Traffic module for an explanation of its use.
- Cost Matrix File (parameter, non-local): Contains the path to the file used to load the Cost Matrix and Original Cost Matrix memories. The path is supplied at the simulation system level.
- Traffic Matrix File (parameter, non-local): Contains the path to the file used to load the Traffic Matrix memory. The path is supplied at the simulation system level.

- Number of Nodes (parameter, non-local): This parameter is instantiated at the simulation system level. It is used here with the read and write matrix primitives to control row and column access. All matrices used in the model are square NxN, where N = the Number of Nodes parameter.
- The arguments Global Average Sum/Count, and Average Delay Window Size are exported from the Compute Global Average module.

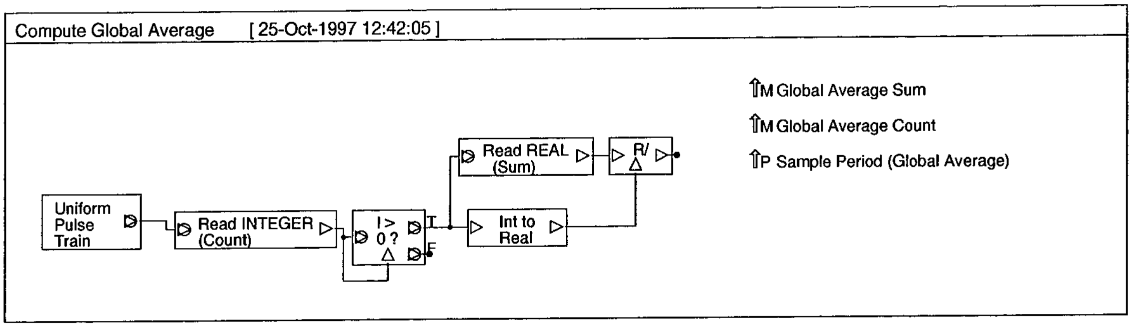


Figure 18. Compute Global Average

The Compute Global Average Module reads the global memories containing the count of all packets that have been delivered to their final destination and the sum of the delays of those packets. The delay of a packet is the time it was delivered to its destination minus the time it was created. The uniform pulse train fires with inter-firing times equal to the window size over which the delays for the network are averaged. The window period is controlled at the simulation system level. A probe can be added to the output port of the R/ block to generate data on packet delays during the simulation. The delay data is global in nature, it is a report on overall network performance.

Parameters:

- Global Average Sum (memory, non-local): Contains the sum of the delay of every packet that has been delivered to its final destination during the simulation of the system. It is computed at the network layer of all nodes in the network.

- Global Average Count (memory, non-local): Contains the count of every packet that has been delivered to its final destination during the simulation of the system. It is computed at the network layer of all nodes in the network.
- Sample Period (Global Average) (parameter, non-local): This parameter is renamed Average Delay Window Size in the Init Network module. It is instantiated at the simulation system level and controls the firing of the Uniform Pulse Train primitive within the module. Every time the Uniform Pulse Train fires, a new value is output from the R/ block.

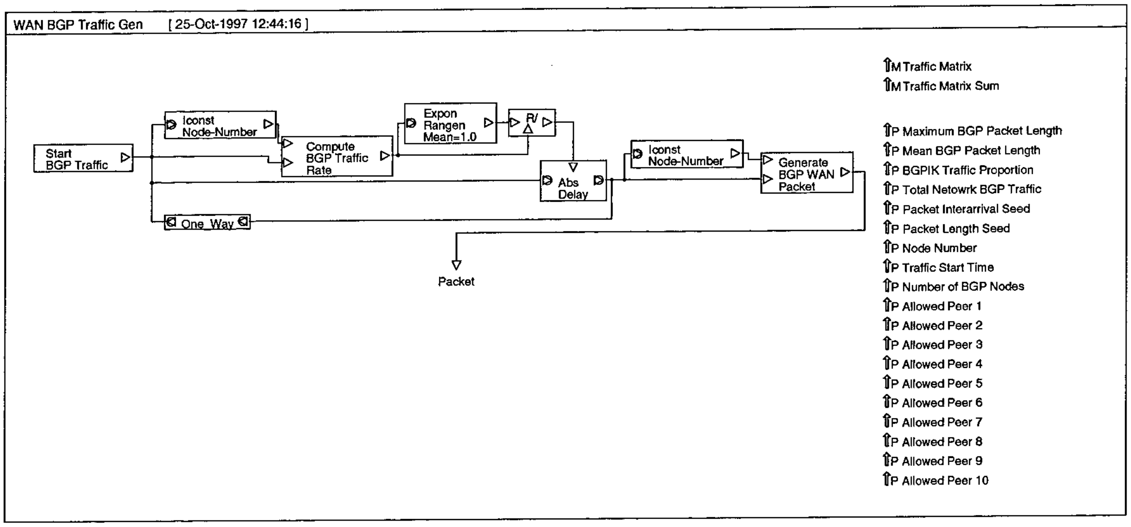


Figure 19. WAN BGP Traffic Generator

This module accepts node number pairs in the form of {this node number, remote node number} from the Iconst Node-Number and Start BGP Traffic blocks respectively. The remote node number can be any external BGP node in the network. Note that the BGP node IDs in this model are 1 through 14 inclusive. These pairs are fed into the Compute BGP Traffic Rate block. The output of this block is the a relative traffic rate between the node pair given as input. This result Is then divided into the output of the Expon Rangen Mean=1.0 block to get an exponentially generated packet interarrival rate in packets/second. The Abs Delay block serves as the data structure that enforces the interarrival time output from the R/ block. After the computed interarrival time the Abs Delay Block fires allowing the packet to be generated. The traffic rate between any two nodes is given by Eq (9). The arguments in Eq (9) are named the same as the parameters discussed in Figure 17. Note that data-type traffic is generated by the WAN Traffic Generator (not shown). The WAN Traffic Generator is the same as this module except that it: a) outputs data traffic, b) generates traffic to all nodes, not only nodes designated as BGP nodes, and c) uses the parameter Total Network Traffic to compute its traffic rate.

Parameters:

- Traffic Matrix (memory, non-local): Used in the computation of interarrival times for packets generated at this node and destined for any BGP node.
- Traffic Matrix Sum (memory, non-local): Used in the computation of interarrival times for packets generated at this node and destined for any BGP node.
- Maximum BGP Packet Length (parameter, non-local): The maximum size of a BGP packet in bits. It is equal to the maximum data packet size. This parameter is instantiated at the simulation system level.
- Mean BGP Packet Length (parameter, non-local): The mean size of a BGP packet in bits. It is equal to the mean data packet size. This parameter is instantiated at the simulation system level.
- BGPIK Traffic Proportion (parameter, non-local): This contains a string value that is the path of the file that is used to control the distribution of BGP Interim Update (BGPI) messages to BGP Keepalive (BGPK) messages. It is instantiated at simulation system level.
- Total Network BGP Traffic (parameter, non-local): This is instantiated at the simulation system level and is equal to a percentage of the Total Network Traffic [14]. This

percentage is based on the nominal model operation and will not vary as traffic loads are modeled within the same simulation model.

- Packet Interarrival Seed (parameter, non-local): Seeds are used as input to random number generators. All seeds in this model are set equal to -1. This allows the Global Seed, which is instantiated at the simulation system level, to be used.
- Packet Length Seed (parameter, non-local): See explanation for Packet Interarrival Seed.
- Node Number (parameter, non-local): The number of the current node. It is used in this module as input to the Compute BGP Traffic Rate module (explained above), and the Generate BGP WAN Packet module where it becomes the Source Host and Source IMP fields of the packet.
- Traffic Start Time (parameter, non-local): This is instantiated at the simulation system level and controls when all traffic generation begins in the model.
- Number of BGP Nodes (parameter, non-local): The value for this parameter is 14 and is instantiated at the simulation system level. It is used in the Start BGP Traffic module.
- Allowed Peer 1 .. 10 (parameter, non-local): This parameter is instantiated at the Node module level. Since traffic is generated for any BGP node indiscriminately (see the Start BGP Traffic module), this parameter controls the flow of BGP traffic to only the allowed peers within the model configuration.

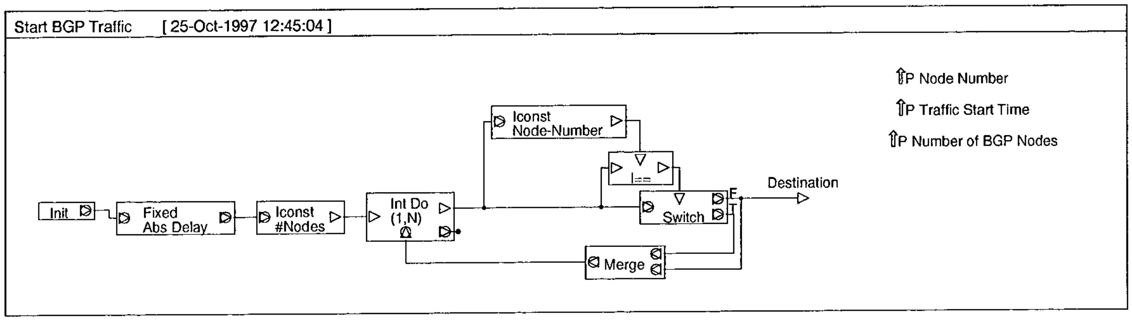


Figure 20. Start BGP Traffic

The function of this module is to supply a continuous stream of destination node numbers to the WAN BGP Traffic Generation module. The Init block fires automatically at simulation start time. The Fixed Abs Delay block delays traffic generation by the Traffic Start Time parameter to ensure that the Init Network module has finished. The Number of BGP Nodes parameter is output from the Iconst #Nodes block and is fed into the Int Do (1,N) block which outputs a steady stream of integers representing the destination node IDs for the BGP traffic. The loopback to the Int Do (1,N) block ensures that the stream will continue indefinitely (the loop will continue for as long as the simulation). The only integer that this module will not output is the current Node Number. That is, no packet will be generated from a node that where the Source Host equals the Destination Host fields of the packet.

Parameters:

- Node Number (parameter, non-local): This parameter is instantiated in the Node module (either the WAN Node, or the BGP WAN Node). Its use in this module is explained above.
- Traffic Start Time (parameter, non-local). This parameter is instantiated at the simulation system level. Its use in this module is explained above.
- Number of BGP Nodes (parameter, non-local). This parameter is instantiated at the simulation system level. Its use in this module is explained above.

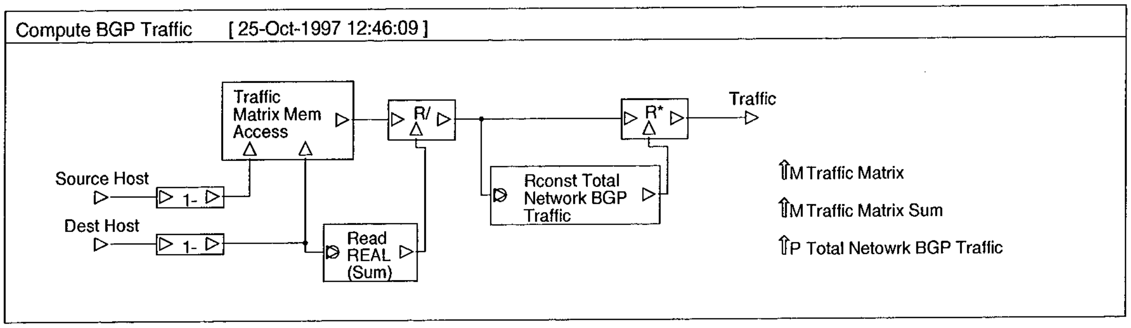
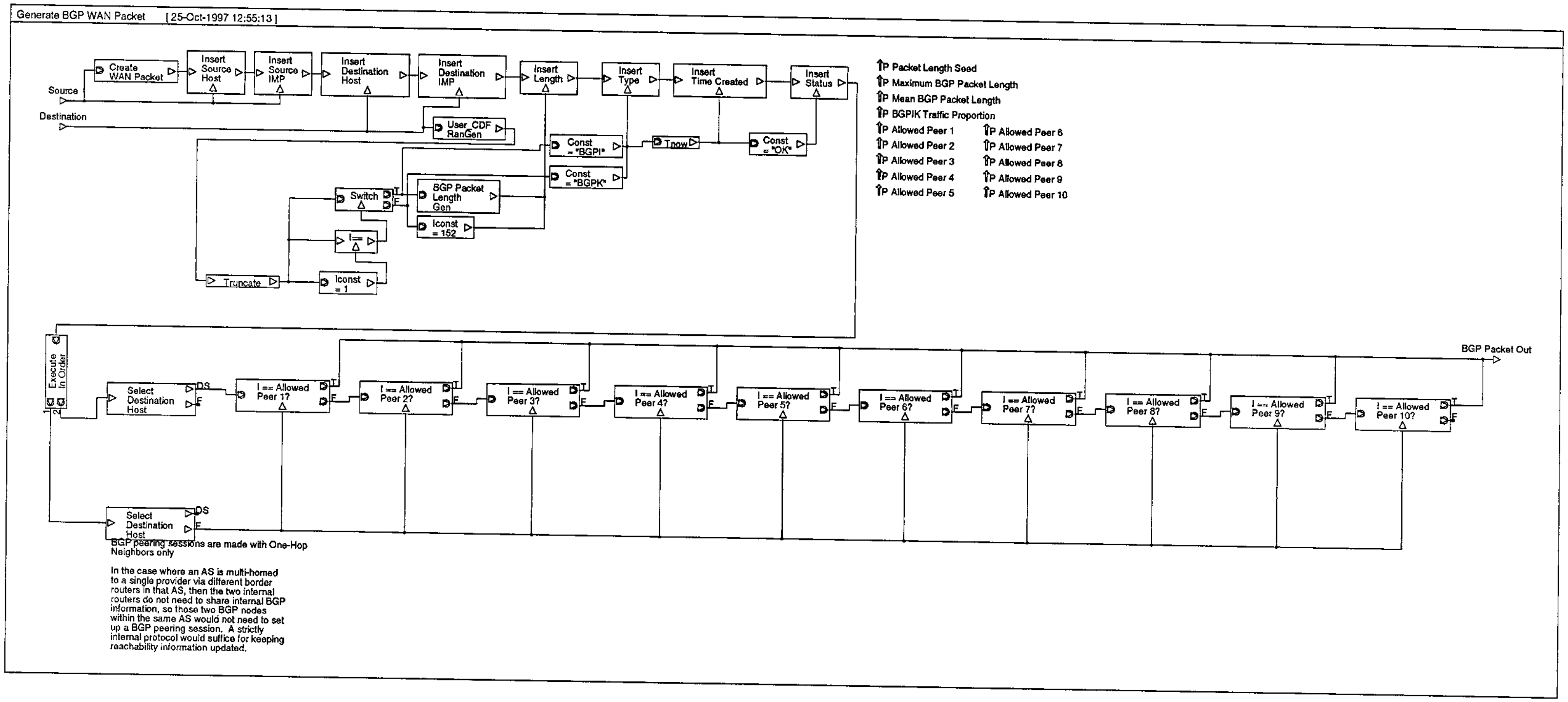

Compute BGP Traffic [25-Oct-1997 12:46:09]
Traffic Matrix Mem Access
R/
R*
Traffic
Source Host
1-
Dest Host
1-
Read REAL (Sum)
Rconst Total Network BGP Traffic
⇑M Traffic Matrix
⇑M Traffic Matrix Sum
⇑P Total Netowrk BGP Traffic

Figure 21. Compute BGP Traffic

This module is responsible for performing all of Eq (9) except the division into ≅1. The BONeS Designer matrix data structures are zero-based and the model's <u>Node Number</u>(s) are 1-based. That is why the row and column indices (the Source Host and Destination Host) are decremented by one when before being fed to the Traffic Matrix Mem Access module.

Parameters: (all parameters are explained at Figure10).



BGP peering sessions are made with One-Hop Neighbors only

In the case where an AS is multi-homed to a single provider via different border routers in that AS, then the two internal routers do not need to share internal BGP information, so those two BGP nodes within the same AS would not need to set up a BGP peering session. A strictly internal protocol would suffice for keeping reachability information updated.

Figure 22. Generate BGP WAN Packet

When a BGP WAN packet is generated, all of its fields (see Table 8), with the exception of Tx Start Time, are initialized. The Length field is a constant 152 bits if the packet represents a BGPK message. If the packet is a BGPI message, its length is exponentially generated by a random number generator with a mean of 1120 bits. Since this module can receive a Destination Host id of any number corresponding to a BGP node number, the destination id is checked against an allowed peer list (represented by Allowed Peer 1 .. 10). In this model, as in the real internet, there is not a full mesh of BGP peering sessions between routers. The local policy at each border router within each AS determines the topology of the peering sessions. The exception is that the core (or top-level) routers must be connected by a full mesh.

Parameters: (all parameters are explained at Figure 19).

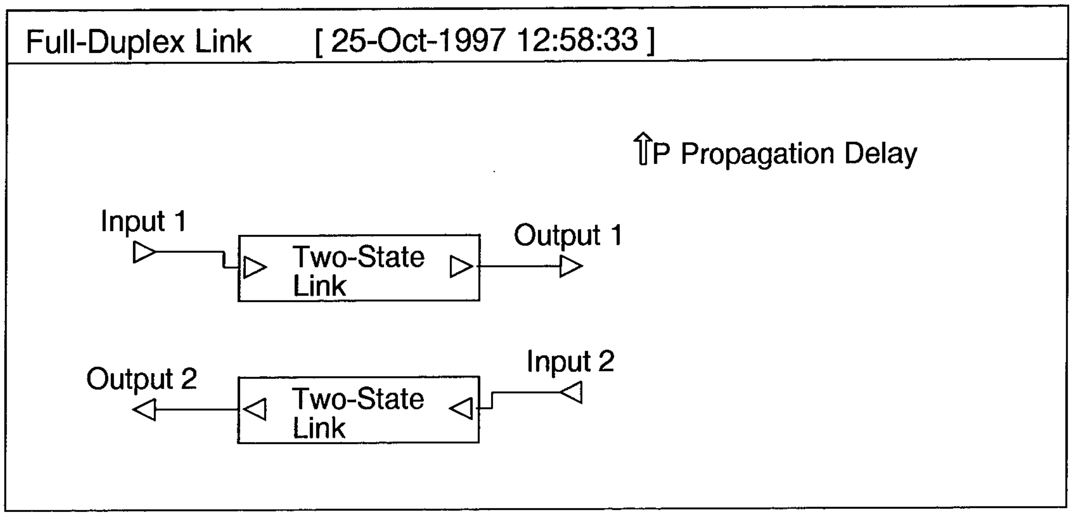


Figure 23. Full Duplex Link

The physical layer representation in the model consists of the Full Duplex Link. It transmits traffic in both directions simultaneously. Both of the data streams together are used in the link utilization computation.

Parameters:

- Propagation Delay (parameter, non-local): This parameter is instantiated at the Node level. Its value is given by Eq (3). All WAN packets entering the Two State Link undergo an absolute delay equal to Propagation Delay.

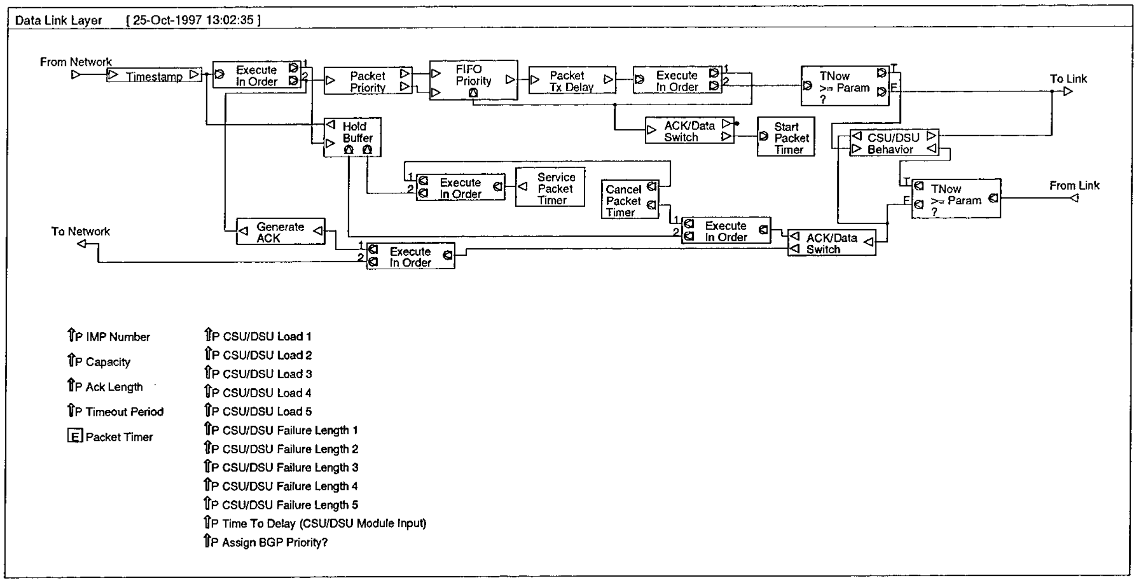


Figure 24. Data Link Layer

The data link layer models packet transmission delay. The transmission queue for the node is also contained here and is another source of delay. A third source for delay is the CSU/DSU Behavior module where packets get delayed if the link utilization is above a user-defined threshold. Both the packets from the host node network layer and the remote node link are fed into the CSU/DSU Behavior module after a user-defined amount of simulation time (Time To Delay (CSU/DSU Module Input)). The delay in routing packets through this module is purposefully assigned to allow simulation start-up transients to die out before link utilization is measured. Packets entering the Data Link Layer module from the link level are acknowledged. The ACK packets are transmitted with priority from the packet queue. BGP packets can also be transmitted with priority if the Assign BGP Priority? parameter is set to “Yes”. All packets leaving this module for the remote host are queried by the ACK/Data switch. If the packet is a data packet, then the ACK timer is started. If no ACK is received from the remote node within the Timeout Period, then the Service Packet Timer module executes allowing the copy of the original data packet, which is in the Hold Buffer, to be retransmitted. If the Service Packet Timer module does not execute this means that an ACK packet was received in time and the copy of the original data packet, which is being held in the Hold Buffer, is discarded. ACKs received from the remote node cancel the timer that was started when the corresponding data packet was transmitted to that node. BONeS Designer timers are controlled by a unique handle id. These id's are instantiated using a counter. The Start Packet Timer and Cancel Packet Timer modules have counter primitives that are kept in synchronization by the packet flow. The Service Packet Timer module also updates the counter in the Cancel Packet Timer module.

Parameters:

- IMP Number (parameter, non-local): This parameter is used by the Timestamp module. IMP stands for Interim Message Processor and is equal to the current node's Node Number. It is inserted in the Source IMP field to allow the modules at the next node's network layer to make the proper routing decision for the packet.
- Capacity (parameter, non-local): This parameter is used in the CSU/DSU Behavior module to as an input to the Throughput primitive. It is instantiated in the Node module.
- Ack Length (parameter, non-local): Instantiated at the simulation system level as 92 bits.
- Timeout Period (parameter, non-local): This parameter is instantiated at the Node module. See Eq (1) for an explanation.
- Packet Timer (event, local): The event associated with the timer modules within the Data Link Layer. Much of the functionality of events are abstracted from the model designer in BONeS and are processed internally. For instance, the Service Packet Timer module does not have a "handle" id. The association of this module to its appropriate Start/Cancel Packet Timer objects is kept internally by BONeS.

- CSU/DSU Load 1/ … /5 (parameter, non-local): These parameters are instantiated in the Node module. See the explanation of this parameter in chapter 4, section: “Instantiation of Particular Model Parameters”.
- CSU/DSU Failure Length 1/ … /5 (parameter, non-local): These parameters are instantiated in the Node module. See the explanation of this parameter in chapter 4, section: “Instantiation of Particular Model Parameters”.
- Time To Delay (CSU/DSU Module Input (parameter, non-local): This parameter is instantiated at the simulation system level. It is used to allow simulation startup transients to die out before packets are allowed to enter the CSU/DSU Behavior module where link utilization is computed.
- Assign BGP Priority? (parameter, non-local): This parameter is instantiated at the simulation system level. If set to yes, then all BGP traffic in the model gets the same transmission priority as ACK packets do. This parameter is meant to be set for the duration of the simulation. Along with traffic rate and hashing BGP traffic, it is one of the main variables of the thesis.

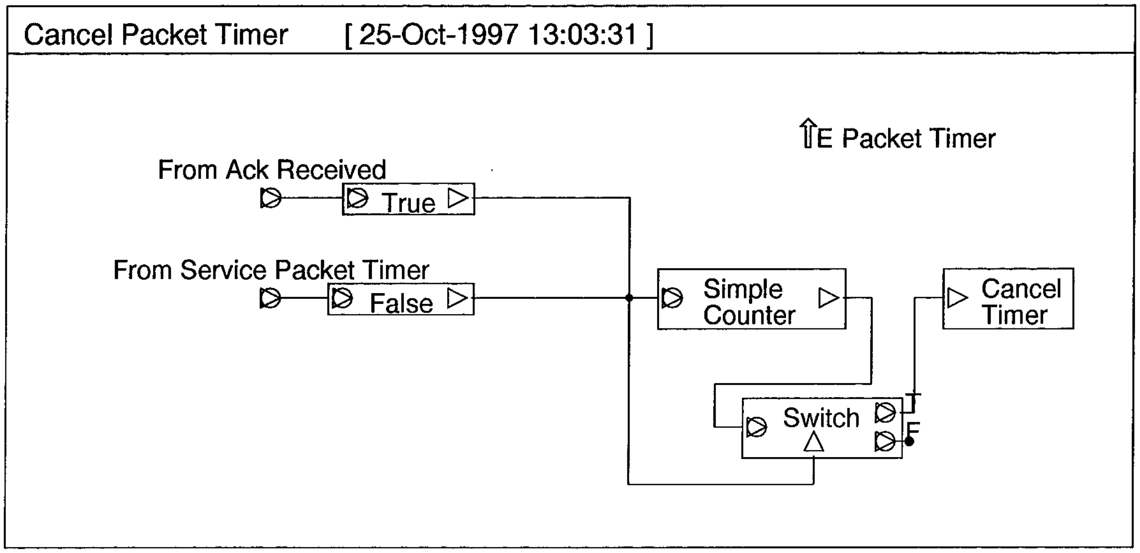


Figure 25. Cancel Packet Timer

This module operate similarly to the Start Packet Timer module. The Cancel Packet Timer module is activated by either the receipt of an ACK packet or the execution of the Service Packet Timer module. This allows the counter to be updated so that both counters, one in the Start Packet Timer module and its corresponding counter in the Cancel Packet Timer module can stay synchronized. The execution of the Service Packet Timer module automatically cancels the associated timer, therefore, in this module, the signal received from the Service Packet Timer event is not routed to the Cancel Timer primitive.

Parameters:

- Packet Timer (event, non-local): Explained above.

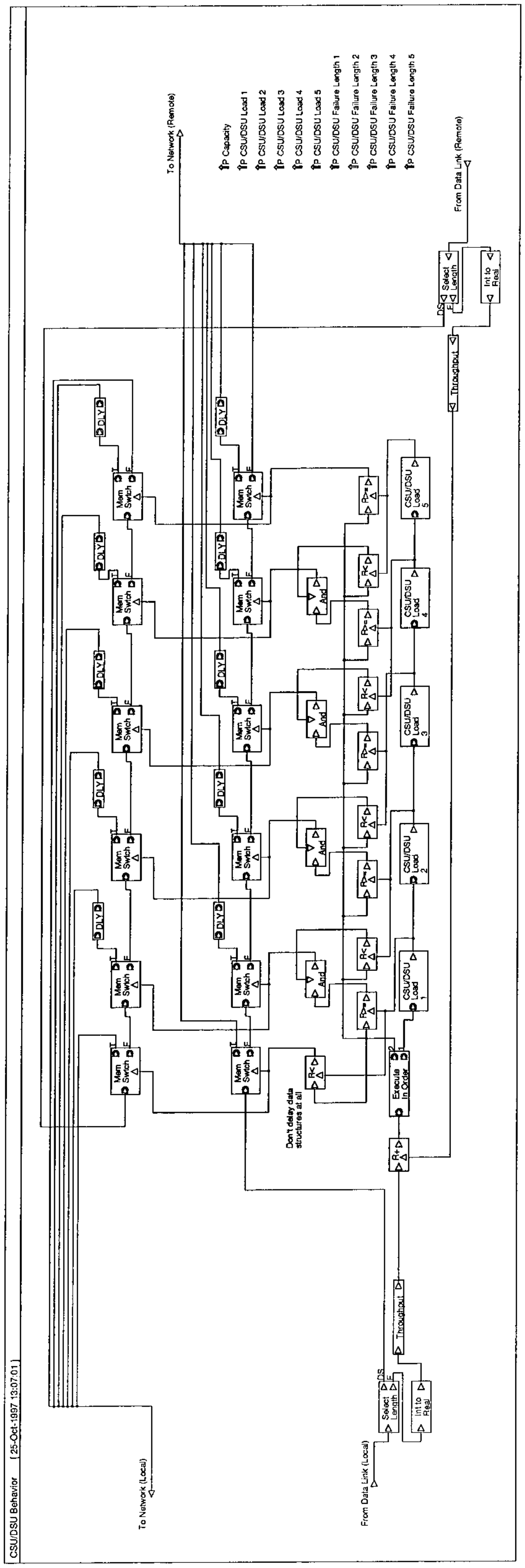


Figure 26. CSU/DSU Behavior

All packets entering this module are delayed if the current link utilization falls within 5 user-defined bins. This behavior is meant to mimic the delay cause by retransmitting packets on a link in which the congestion is causing retransmission requests. See Eq (5) for a detailed explanation of the delay times. The packet is not delayed at all if the current link utilization is below the user-defined threshold (this is the first decision made on the packet when entering the module). The Length fields of both the incoming and outgoing packets are used to compute current link utilization. Note that the BONeS-supplied Throughput primitive operates from a Uniform Pulse Train figure. This causes non-active links in the model to emit divide by zero errors. Therefore Capacity parameter of non-active links in the model have to be set to "1". This is trivial and doesn't not affect the correctness of model operation.

Parameters:

- Capacity (parameter, non-local): The capacity in bits/second of the current link. This parameter is instantiated in the Node module.
- CSU/DSU Load 1/ … /5 (parameters, non-local): Explained in the "Instantiation of Particular Model Parameters" section of chapter 4. Instantiated in the Node module.
- CSU/DSU Failure Length 1/ … /5 (parameters, non-local). Same as above.

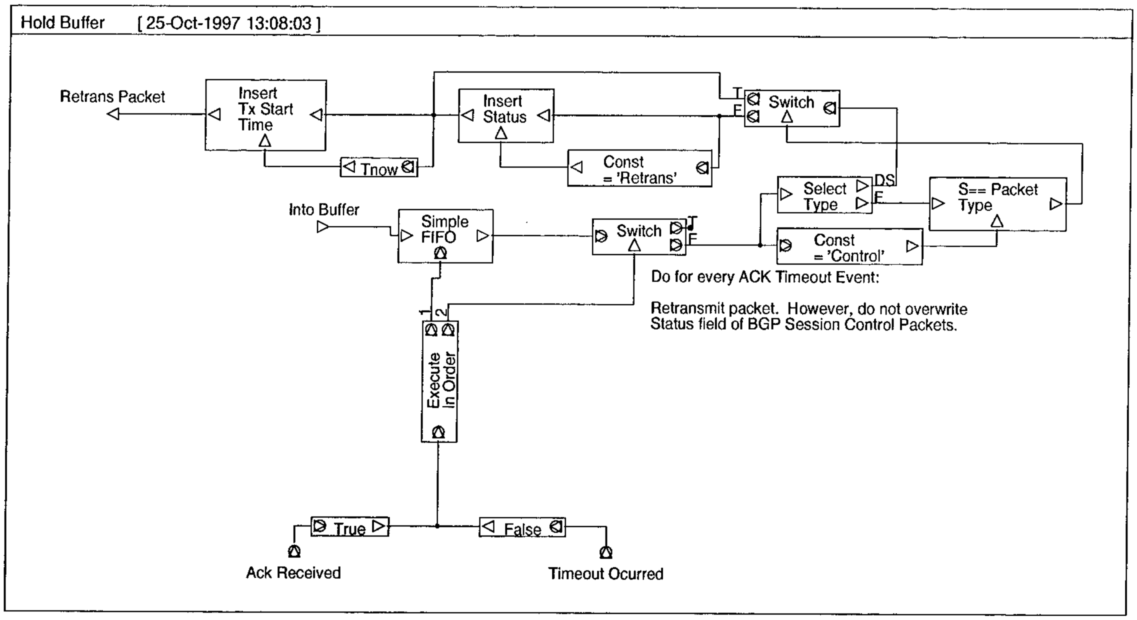


Figure 27. Hold Buffer

All packets leaving the Data Link module from the host node are copied into this buffer. If the corresponding ACK packet is received in within the Timeout Period then the packet is sunk (the T[rue] branch of the Switch that is attached to the Simple FIFO queue is taken), if not the packet is retransmitted on the link. When the packet is retransmitted, its status is changed to "Retrans", unless the packet being retransmitted is a BGP session control packet, and a new time of transmission is inserted into the Tx Start Time field of the packet.

Parameters: None

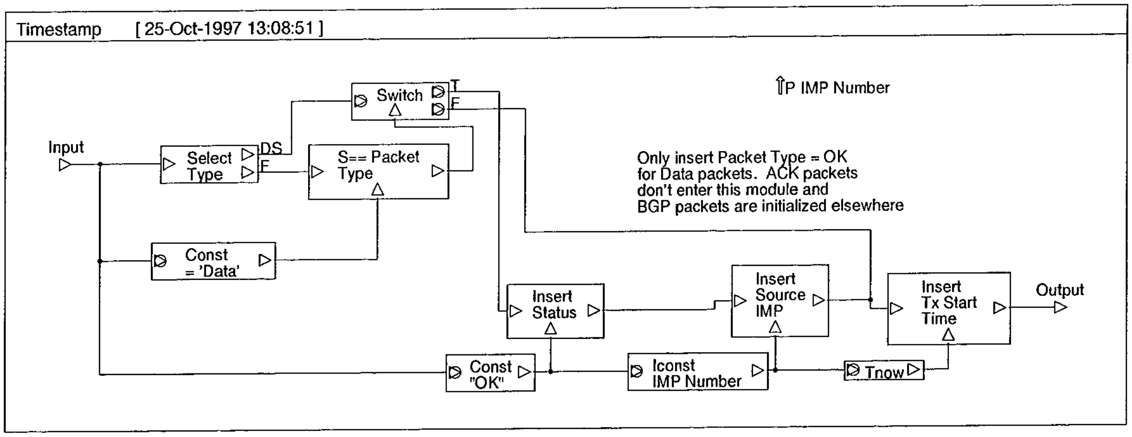
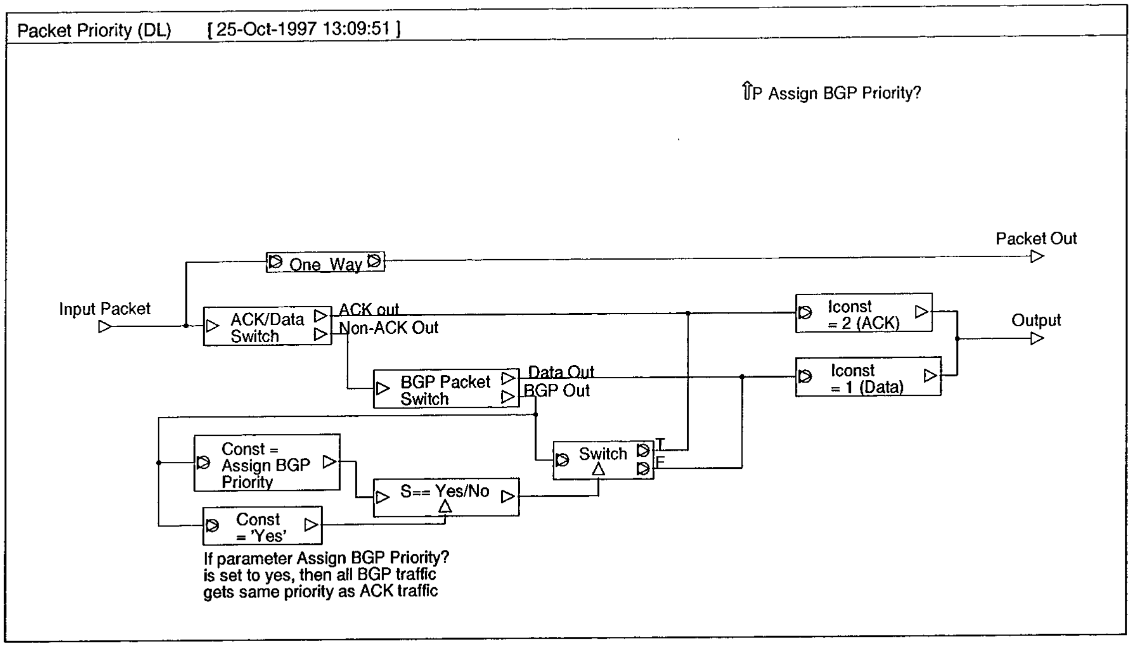

Timestamp [25-Oct-1997 13:08:51]
Input
Select Type
DS
F
Switch
T
F
S== Packet Type
Const = 'Data'
P IMP Number
Only insert Packet Type = OK for Data packets. ACK packets don't enter this module and BGP packets are initialized elsewhere
Insert Status
Insert Source IMP
Insert Tx Start Time
Output
Const "OK"
Iconst IMP Number
Tnow

Figure 28. Timestamp

Data packets entering this module have their Status, Source IMP, and Tx Start Time fields updated. BGP packets just have their Tx Start Time Field updated as the other fields are initialized in the WAN BGP Traffic Generator module.

Parameters:

- IMP Number (parameter, non-local): Is set equal to the current node's Node Number in the Node module. This parameter, along with the Destination Host field, is used at the next hop node's network layer blocks to make a routing decision.



If parameter Assign BGP Priority? is set to yes, then all BGP traffic gets same priority as ACK traffic

Figure 29. Packet Priority

If the incoming packet is a BGP message, it is assigned the same priority as the ACK packets if the Assign BGP Priority? parameter is set to "Yes". If the BGP packet is transmitted with priority, it is moved ahead of all lower priority packets but behind any packet presently being transmitted as the BGP packet enters the queue. If there are packets with the same priority that are queued as a packet enters the queue with priority, then all the packets having priority are treated in FCFS fashion but, as a group, are in front of all other lower-priority packets.

Parameters: see the discussion at Figure 24.

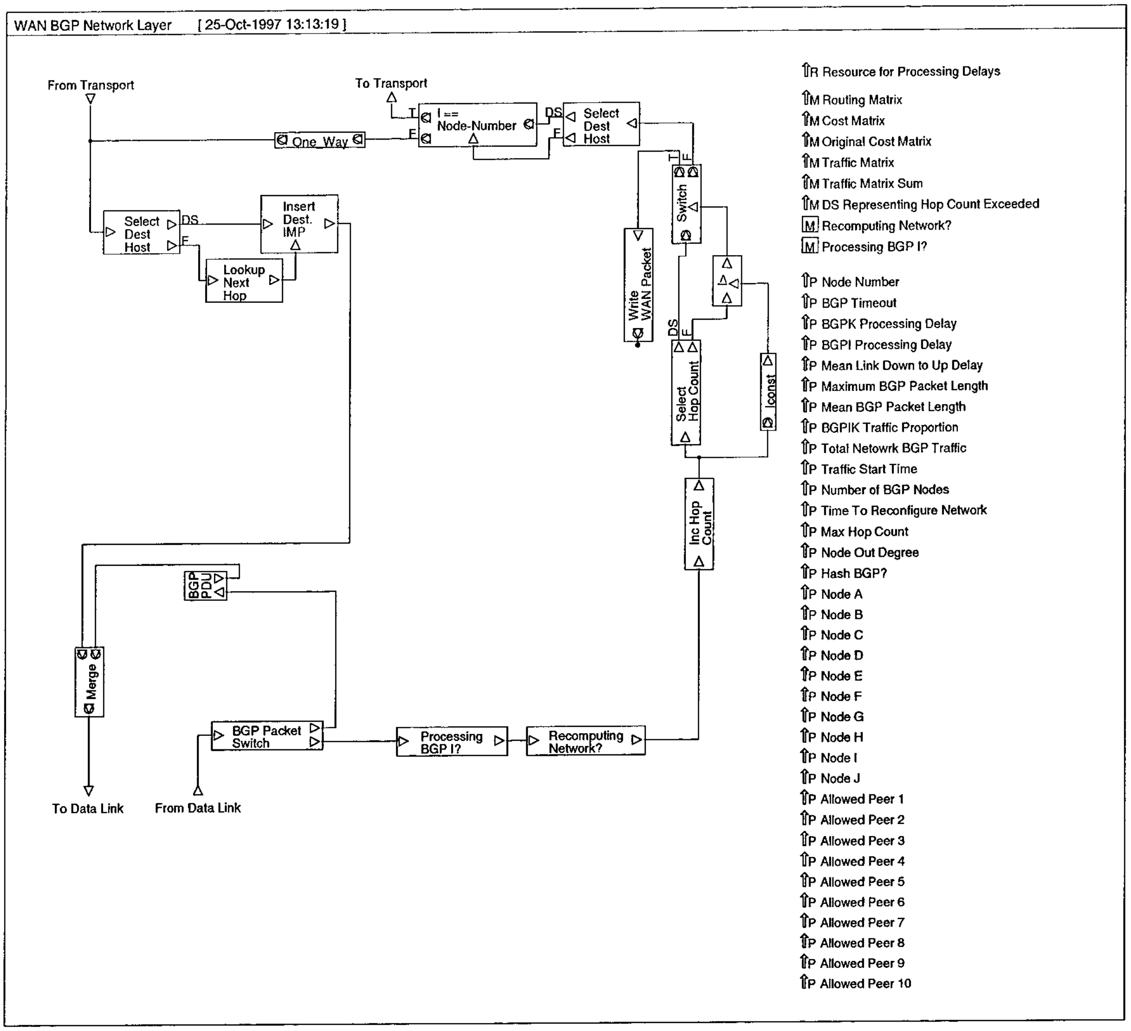


Figure 30. WAN BGP Network Layer

The network layer modules account for the largest part of the Transit-Stub ISM. The WAN BGP Network Layer is identical to the WAN Network Layer with BGP processing capability added indicated by the BGP Packet Switch, Processing BGP I?, Recomputing Network?, and BGP PDU blocks.

WAN data packets (non-BGP packets) entering the module from the data link layer are switched into the Processing BGP I? block where it undergoes a delay if the node is currently processing a BGPI message. The packet is then routed into the Recomputing Network? module where it undergoes a similar delay if the node is currently processing a BGPF message. If the node is performing MD5 hashing on BGP messages, then the delay encountered by data packets is twice the delay normally encountered (see the explanation justifying the choice of this delay time in Chapter 4 at Eqs (7) and (8)). The WAN packet Hop Count field is then incremented and the packet is sunk if the field is greater than 15. If the data packet is destined for this node (WAN host), it is delivered to the transport layer, if not, it is delivered to the routing decision process and then delivered to the data link to be switched to the next hop in the destination path. Data packets entering from the transport layer (simply the traffic generation function (see Figure 44)) are delivered directly to the routing decision process.

WAN BGP packets are routed into the BGP PDU for processing. The output portion of the BGP PDU carries the BGP traffic generated at this node and destined for its peers. Note that the BGP traffic is injected directly into the link and not routed. This is because the operation of the protocol is based on the hop-by-hop routing paradigm used in the internet. BGP peers are one-hop neighbors of each other and the traffic doesn't need to be routed within this model.

Parameters (many of these are explained in detail in chapter 4):

- Resource for Processing Delays (resource, non-local): This represents the work done by the BGP PDU when processing the BGP protocol. This resource argument is also used whenever a link fails and the network has to be recomputed. It is modeled as a queue data structure which receives transactions. These transaction data structures enter the queue with request for processor time (queue delay). If the processor queue is empty the transaction stays in the queue (is processed) for the amount of time of the request, otherwise the transaction queues. The queue represents a single server system without preemption or priority. The Resource for Processing Delays argument groups the modeling of all processing delays into one entity. It is localized at the Node level.
- Routing Matrix (memory, non-local): Used by the Lookup Next Hop module as input to the routing decision.

- DS Representing Hop Count Exceeded (memory, non-local): This is global memory that is written to when any packet in the network is sunk because its hop count was exceeded. This allows a single probe to be used to get information during the simulation runs.
- Recomputing Network? (memory, local): When a link fails because of non-receipt of a BGP protocol message within the BGP Timeout window, or a BGPF message is received, the BGP PDU will set this memory value to yes while the network is being recomputed.
- Processing BGP I? (memory, local): Similar to above but references the nodal processing of BGP interim update (BGPI) messages.
- Node Number (parameter, non-local): The number of the current node. Used by the Lookup Next Hop module the make routing decisions. This parameter is instantiated at the Node level.
- BGP Timeout (parameter, non-local): This is the window period in which, if not BGP message is received, then the link is declared to be down. It is instantiated at the simulation system level.
- BGPK Processing Delay (parameter, non-local): The amount of time it takes the current node to process a BGP keepalive message. This is used as input for the nodal processing delay function which is modeled by the Resource for Processing Delays parameter.
- BGPI Processing Delay (parameter, non-local): similar to above but is a longer amount of time.
- Time to Reconfigure Network (parameter, non-local): similar to above but is a longer amount of time yet.
- Mean Link Down to Up Delay (parameter, non-local): This parameter is dependent on the processing power of the current node. It governs the amount of time that it takes to

re-establish a failed BGP peering session and re-instate a previously failed link into operation.

- Maximum BGP Packet Length (parameter, non-local): Equal to the maximum WAN data packet length. Used by the traffic generation modules to set a ceiling on the output of the exponential packet size generator. Instantiated at simulation system level.
- Mean BGP Packet Length (parameter, non-local): Equal to the mean WAN data packet length. Used by the exponential random number generators within the traffic generation modules to set the packet lengths. Instantiated at simulation system level.
- BGPIK Traffic Proportion (parameter, non-local): Instantiated at the simulation system level, it is a pointer to a file that is input to a cumulative distribution function random number generator that controls the distribution of BGPI to BGPK messages.
- Total Network BGP Traffic (parameter, non-local): Instantiated at the simulation system level as a proportion of Total Network Traffic.
- Traffic Start Time (parameter, non-local): This parameter is instantiated at the simulation system level and controls the beginning of traffic generation in the model. It is given as a small delta to simulation start time.
- Number of BGP Nodes (parameter, non-local): instantiated at the simulation system level, it is used by the Start BGP Traffic module to output appropriate destination BGP node numbers. The BGP Node Number(s) in this model should be consecutively numbered beginning with one.
- Max Hop Count (parameter, non-local): This parameter is instantiated at the simulation system level. It derivation is explained in chapter 4.
- Node Out Degree (parameter, non-local): Instantiated at the Node level. Its use is explained in chapter 4. This parameter could really be thought of as simply node degree as all links are bi-directional.

- Node A ... J (parameter, non-local): These parameters are instantiated at the Node level and are set equal to the Node Number that the current node is attached to on links A ... J. These parameters ensure the correct switching of packets in the network.
- Allowed Peer 1 ... 10 (parameter, non-local): These parameters are instantiated at the Node level and are the Node Number(s) of the allowed BGP peers of the current node. Since BGP traffic is generated to all nodes indiscriminately by the 1 ... N Do Loop block in the Start BGP Traffic module, these parameters are used as a filter to BGP traffic.
- Hash BGP? (parameter, non-local). Instantiated at the simulation system level, this parameter is set to "Yes" or "No" and is set for the length of the simulation. Along with traffic rate, and assigning priority to the BGP protocol messages, it is one of the main variables in this model. If yes, then there is a delay associated with data packets which are traversing the node at which BGP messages are being hashed.

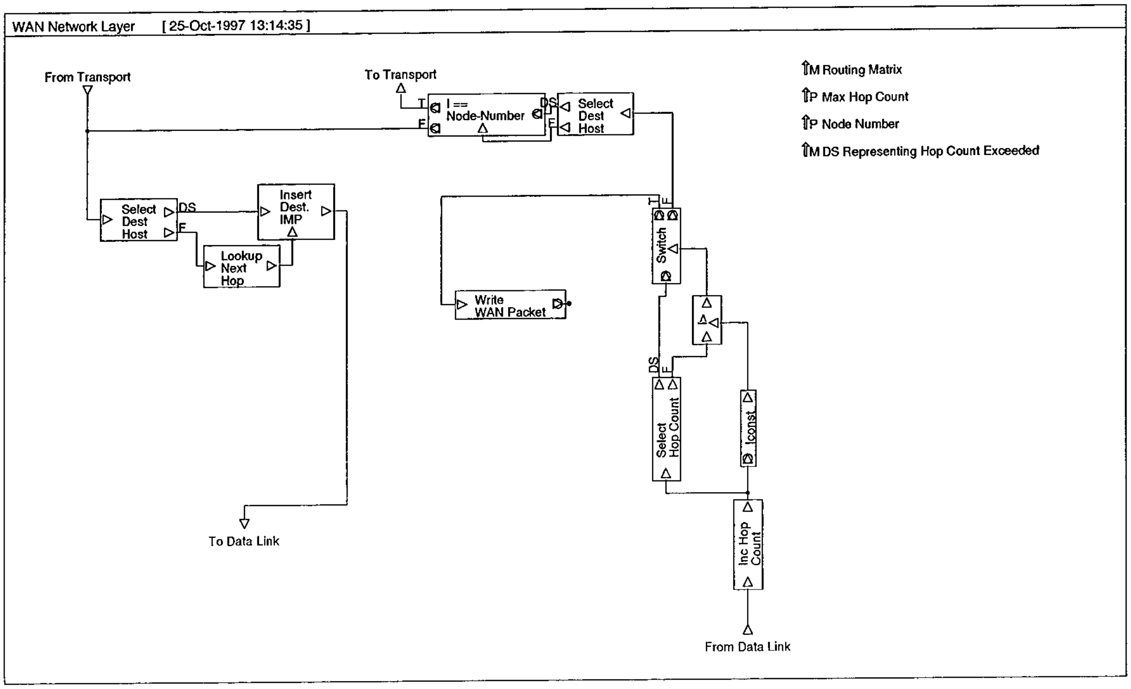
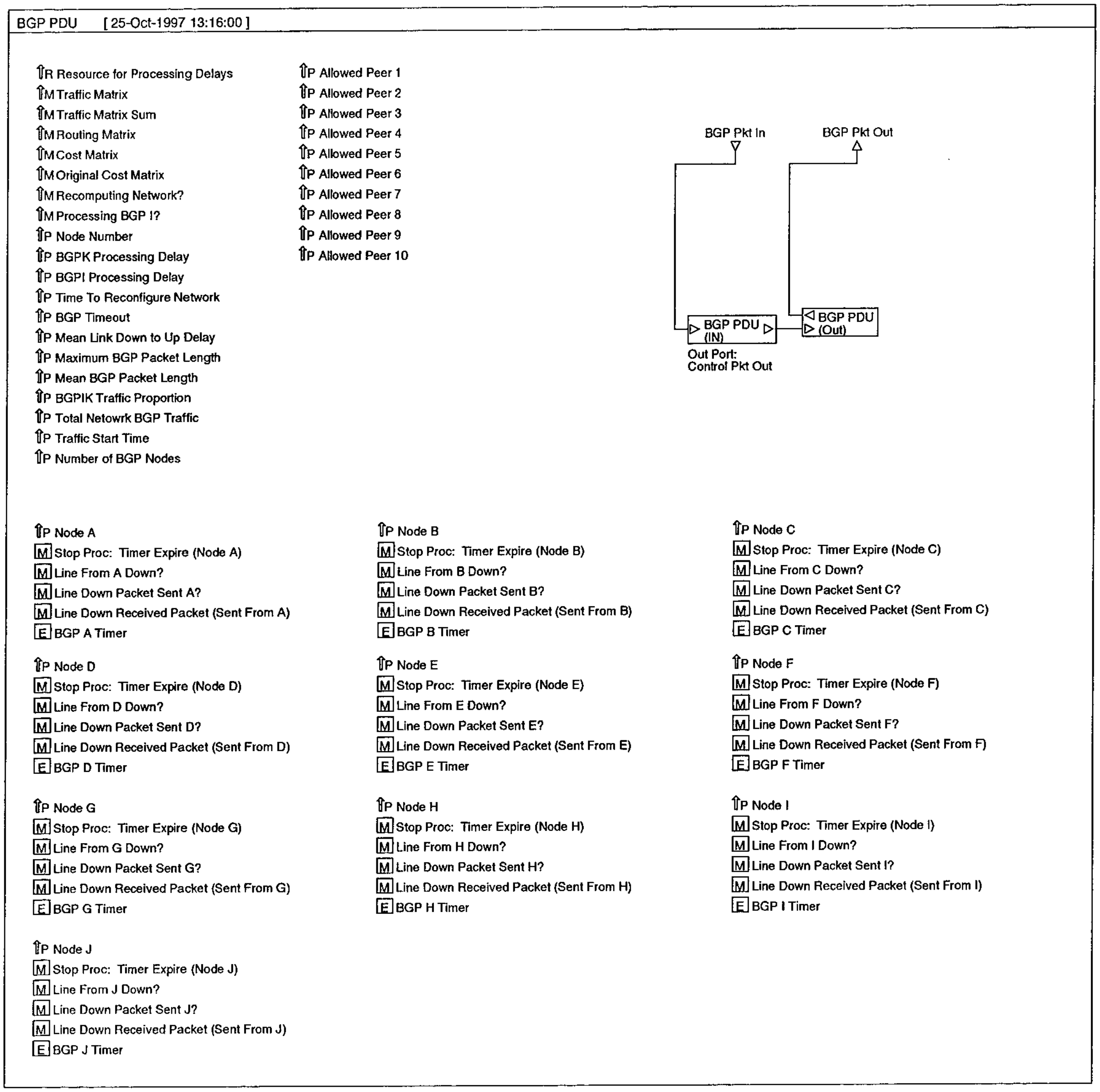


Figure 31. WAN Network Layer

The WAN Network Layer is the same module as the WAN BGP Network Layer without the BGP functionality. All non-BGP nodes in the model will have this module as their network layer. It is shown here for completeness.

Parameters: Discussed above.



BGP PDU [25-Oct-1997 13:16:00]

⇧R Resource for Processing Delays
⇧M Traffic Matrix
⇧M Traffic Matrix Sum
⇧M Routing Matrix
⇧M Cost Matrix
⇧M Original Cost Matrix
⇧M Recomputing Network?
⇧M Processing BGP I?
⇧P Node Number
⇧P BGPK Processing Delay
⇧P BGPI Processing Delay
⇧P Time To Reconfigure Network
⇧P BGP Timeout
⇧P Mean Link Down to Up Delay
⇧P Maximum BGP Packet Length
⇧P Mean BGP Packet Length
⇧P BGPIK Traffic Proportion
⇧P Total Netowrk BGP Traffic
⇧P Traffic Start Time
⇧P Number of BGP Nodes

⇧P Allowed Peer 1
⇧P Allowed Peer 2
⇧P Allowed Peer 3
⇧P Allowed Peer 4
⇧P Allowed Peer 5
⇧P Allowed Peer 6
⇧P Allowed Peer 7
⇧P Allowed Peer 8
⇧P Allowed Peer 9
⇧P Allowed Peer 10

⇧P Node A
🄼 Stop Proc: Timer Expire (Node A)
🄼 Line From A Down?
🄼 Line Down Packet Sent A?
🄼 Line Down Received Packet (Sent From A)
🄴 BGP A Timer

⇧P Node B
🄼 Stop Proc: Timer Expire (Node B)
🄼 Line From B Down?
🄼 Line Down Packet Sent B?
🄼 Line Down Received Packet (Sent From B)
🄴 BGP B Timer

⇧P Node C
🄼 Stop Proc: Timer Expire (Node C)
🄼 Line From C Down?
🄼 Line Down Packet Sent C?
🄼 Line Down Received Packet (Sent From C)
🄴 BGP C Timer

⇧P Node D
🄼 Stop Proc: Timer Expire (Node D)
🄼 Line From D Down?
🄼 Line Down Packet Sent D?
🄼 Line Down Received Packet (Sent From D)
🄴 BGP D Timer

⇧P Node E
🄼 Stop Proc: Timer Expire (Node E)
🄼 Line From E Down?
🄼 Line Down Packet Sent E?
🄼 Line Down Received Packet (Sent From E)
🄴 BGP E Timer

⇧P Node F
🄼 Stop Proc: Timer Expire (Node F)
🄼 Line From F Down?
🄼 Line Down Packet Sent F?
🄼 Line Down Received Packet (Sent From F)
🄴 BGP F Timer

⇧P Node G
🄼 Stop Proc: Timer Expire (Node G)
🄼 Line From G Down?
🄼 Line Down Packet Sent G?
🄼 Line Down Received Packet (Sent From G)
🄴 BGP G Timer

⇧P Node H
🄼 Stop Proc: Timer Expire (Node H)
🄼 Line From H Down?
🄼 Line Down Packet Sent H?
🄼 Line Down Received Packet (Sent From H)
🄴 BGP H Timer

⇧P Node I
🄼 Stop Proc: Timer Expire (Node I)
🄼 Line From I Down?
🄼 Line Down Packet Sent I?
🄼 Line Down Received Packet (Sent From I)
🄴 BGP I Timer

⇧P Node J
🄼 Stop Proc: Timer Expire (Node J)
🄼 Line From J Down?
🄼 Line Down Packet Sent J?
🄼 Line Down Received Packet (Sent From J)
🄴 BGP J Timer

Figure 32. BGP Protocol Data Unit

This is the top level representation of the BGP processing capabilities in the model. In order to better manage the complexity of the modules, the BGP PDU was split into “in” and “out” processing modules. Each manages up to 10 independent BGP peering sessions simultaneously, although in this model, only six simultaneous peering sessions are used. All BGP traffic from any peer enters this module via a BGP packet switch (see Figure 30) at the front end of the network layer.

Parameters: The parameters in the top left corner above are explained in Figure 30. The others are for managing the BGP sessions.

- Node Z (parameter, non-local): The Node Number of the remote node connected to the present node over data link A. Used at the Node level to switch packets to the proper output port.
- Stop Proc: Timer Expire (Node Z) (memory, local): Used to control BGP session negotiation between two peers.
- Line From Z Down? (memory, local): Used to control BGP session traffic between peers. If this memory is yes, then the peering session is currently down and BGP traffic bound for the remote node is sunk.
- Line Down Packet Sent Z? (memory, local): Used to control BGP session negotiation between two peers.
- Line Down Received Packet (Sent From Z) (memory, local): Used to control BGP session negotiation between two peers.
- BGP Z Timer (event, local): The BGP session timer. If BGP packets are not received within the BGP Timeout period, then a timer expiration event associated with this parent event occurs.

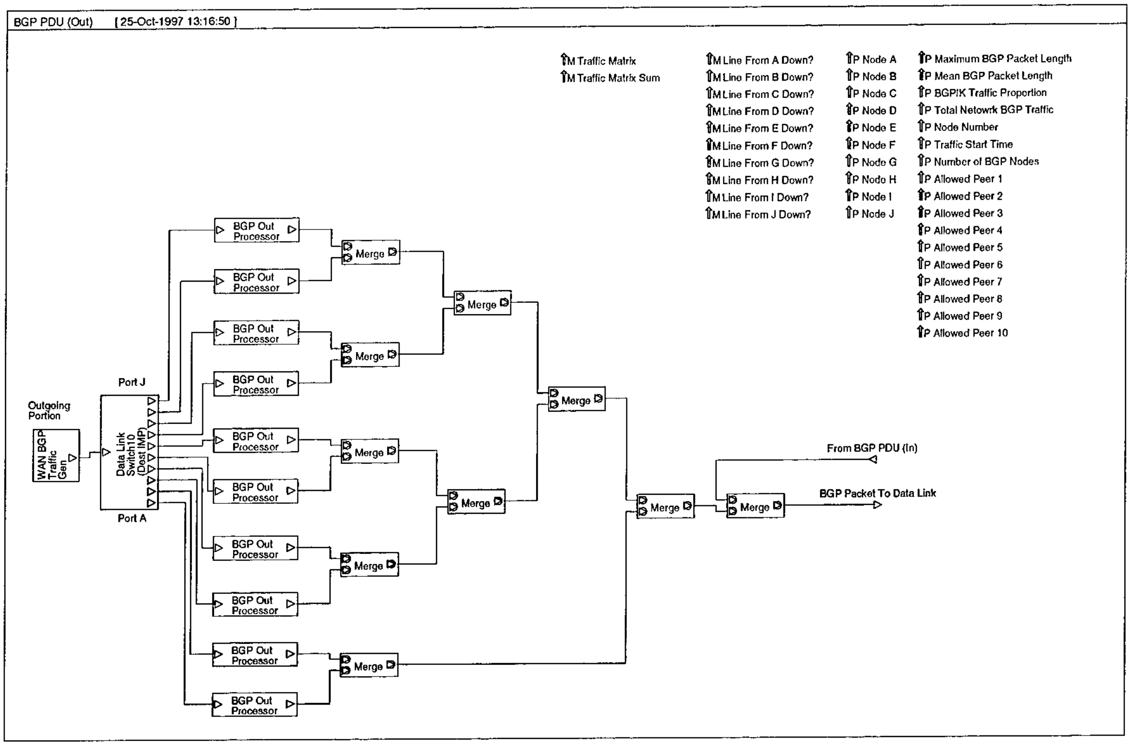
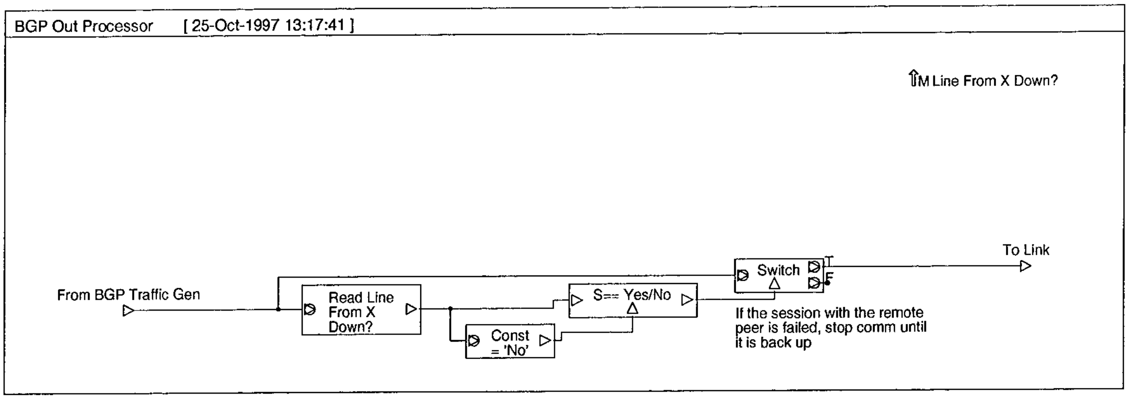


Figure 31. BGP Protocol Data Unit (Out)

This module is responsible for generating BGP traffic in the model. The packet switch examines the destination Node Number of all packets and switches them into the appropriate BGP Out Processor. This module also accepts BGP session control packets from the corresponding BGP PDU (In) module, (see Figure 32), and transmits then over the outgoing network port (see Figure 30).

Parameters: All parameters are explained at Figure 30 with the exception of the memory arguments Line From Z Down? which is explained below.



If the session with the remote peer is failed, stop comm until it is back up

Figure 34. BGP Out Processor

The sole job of the BGP out processor is to sink all BGP traffic with peers whose session with the current node is down. Note that in Figure 33, the outgoing BGP traffic received from the BGP PDU (In) module is not subject to this check. That is because the only outgoing traffic generated from the BGP PDU (In) module is BGP session control traffic. The traffic sunk here are normal BGP update/keepalive messages.

Parameters:

- Line From X Down? (memory, non-local): The type of this memory is Yes/No. This memory is instantiated with A … J in the parent module (see Figure 33). Later, these memory parameters are tied to real nodes when Node modules are added at the simulation system layer. If this memory is "Yes", then outgoing traffic for the particular destination host is sunk.

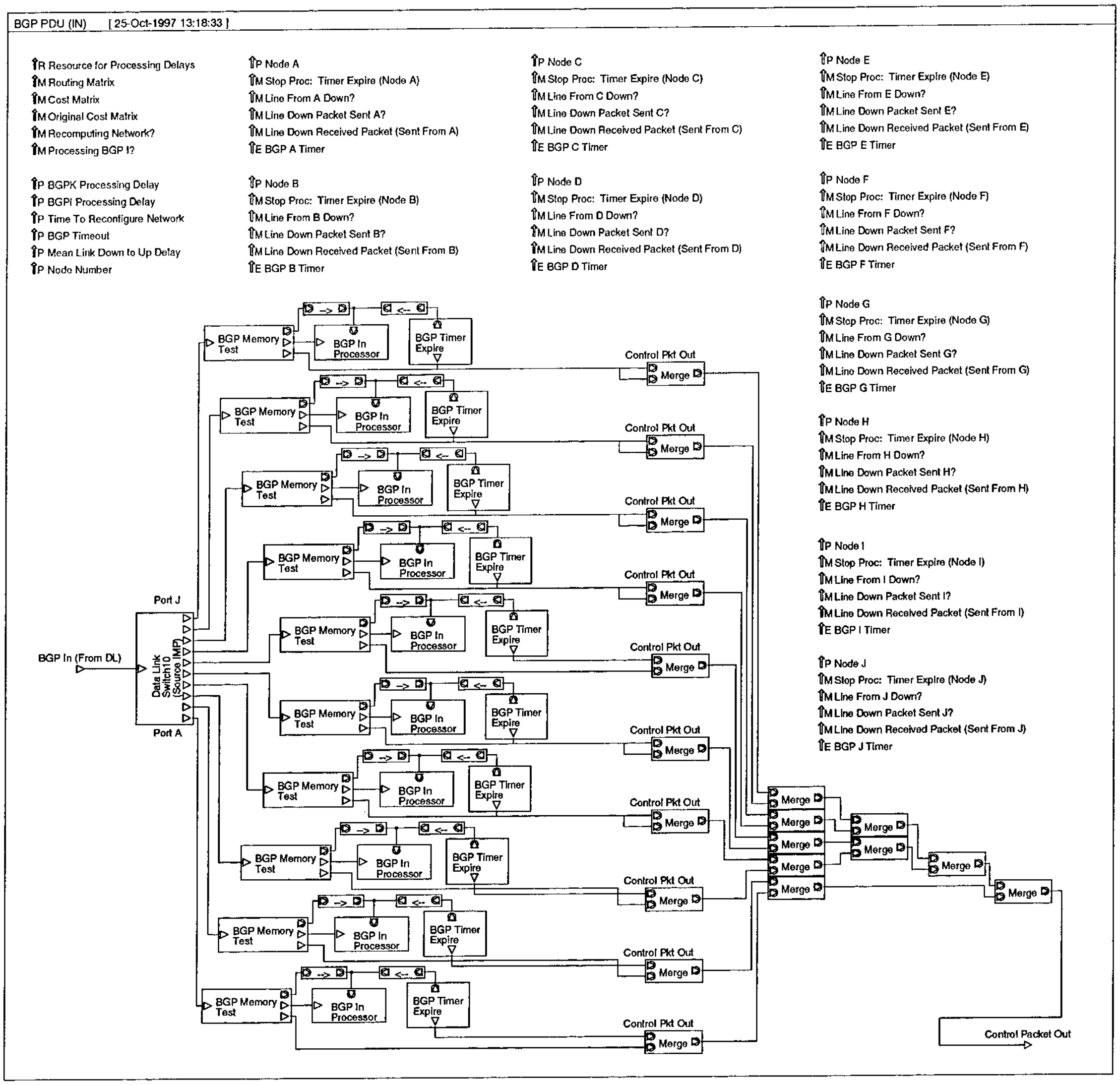
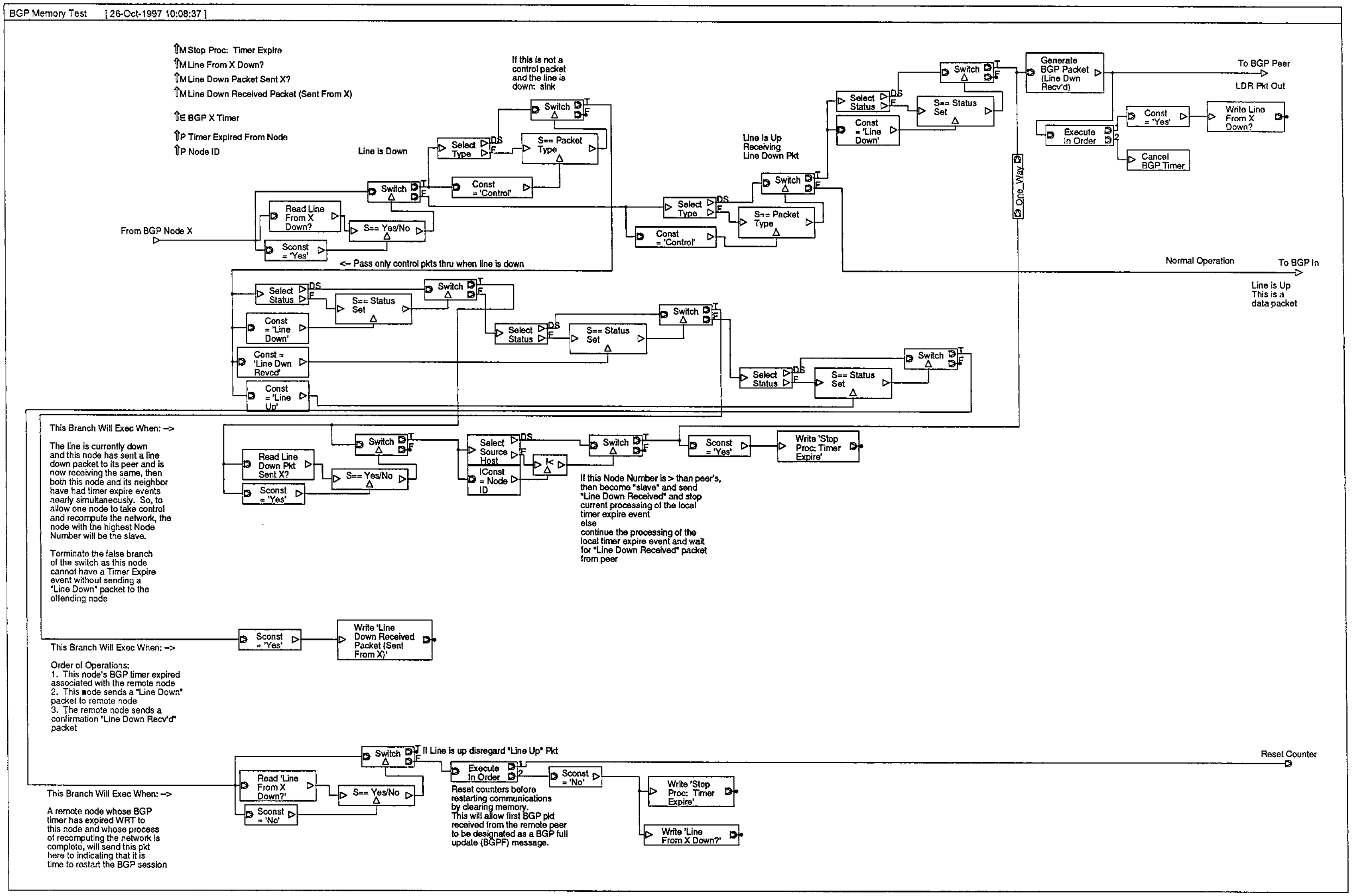

BGP PDU (IN) [25-Oct-1997 13:18:33]
R Resource for Processing Delays
M Routing Matrix
M Cost Matrix
M Original Cost Matrix
M Recomputing Network?
M Processing BGP I?
P BGPK Processing Delay
P BGPI Processing Delay
P Time To Reconfigure Network
P BGP Timeout
P Mean Link Down to Up Delay
P Node Number
P Node A
M Stop Proc: Timer Expire (Node A)
M Line From A Down?
M Line Down Packet Sent A?
M Line Down Received Packet (Sent From A)
E BGP A Timer
P Node B
M Stop Proc: Timer Expire (Node B)
M Line From B Down?
M Line Down Packet Sent B?
M Line Down Received Packet (Sent From B)
E BGP B Timer
P Node C
M Stop Proc: Timer Expire (Node C)
M Line From C Down?
M Line Down Packet Sent C?
M Line Down Received Packet (Sent From C)
E BGP C Timer
P Node D
M Stop Proc: Timer Expire (Node D)
M Line From D Down?
M Line Down Packet Sent D?
M Line Down Received Packet (Sent From D)
E BGP D Timer
P Node E
M Stop Proc: Timer Expire (Node E)
M Line From E Down?
M Line Down Packet Sent E?
M Line Down Received Packet (Sent From E)
E BGP E Timer
P Node F
M Stop Proc: Timer Expire (Node F)
M Line From F Down?
M Line Down Packet Sent F?
M Line Down Received Packet (Sent From F)
E BGP F Timer
P Node G
M Stop Proc: Timer Expire (Node G)
M Line From G Down?
M Line Down Packet Sent G?
M Line Down Received Packet (Sent From G)
E BGP G Timer
P Node H
M Stop Proc: Timer Expire (Node H)
M Line From H Down?
M Line Down Packet Sent H?
M Line Down Received Packet (Sent From H)
E BGP H Timer
P Node I
M Stop Proc: Timer Expire (Node I)
M Line From I Down?
M Line Down Packet Sent I?
M Line Down Received Packet (Sent From I)
E BGP I Timer
P Node J
M Stop Proc: Timer Expire (Node J)
M Line From J Down?
M Line Down Packet Sent J?
M Line Down Received Packet (Sent From J)
E BGP J Timer
BGP Memory Test
BGP In Processor
BGP Timer Expire
Control Pkt Out
Merge
Port J
Port A
BGP In (From DL)
Data Link Switch10 (Source IMP)
Control Packet Out

Figure 35. BGP Protocol Data Unit (In)

The bulk of BGP protocol processing is accomplished within this module. It accepts all BGP traffic from its peers and switches it to the appropriate bank of modules based on the source Node Number of the incoming BGP packet. The reason that the module is constructed in parallel is because the range of access to and control over timer events afforded by the BONeS Designer software is constricted to primitive operations. In order to keep the events segregated (i.e., support simultaneous BGP peering sessions), it was necessary to duplicate functionality in this module. BGP packets are processed according to their type and then retired. The module also handles all BGP session negotiation between peers.

Parameters: See the following four figures for explanation.



If this is not a control packet and the line is down: sink

This Branch Will Exec When: -->

The line is currently down and this node has sent a line down packet to its peer and is now receiving the same, then both this node and its neighbor have had timer expire events nearly simultaneously. So, to allow one node to take control and recompute the network, the node with the highest Node Number will be the slave.

Terminate the false branch of the switch as this node cannot have a Timer Expire event without sending a "Line Down" packet to the offending node

If this Node Number is > than peer's, then become "slave" and send "Line Down Received" and stop current processing of the local timer expire event
else
continue the processing of the local timer expire event and wait for "Line Down Received" packet from peer

This Branch Will Exec When: -->

Order of Operations:
1. This node's BGP timer expired associated with the remote node
2. This node sends a "Line Down" packet to remote node
3. The remote node sends a confirmation "Line Down Recv'd" packet

If Line is up disregard "Line Up" Pkt

Reset counters before restarting communications by clearing memory.
This will allow first BGP pkt received from the remote peer to be designated as a BGP full update (BGPF) message.

This Branch Will Exec When: -->

A remote node whose BGP timer has expired WRT to this node and whose process of recomputing the network is complete, will send this pkt here to indicating that it is time to restart the BGP session

Figure 36. BGP Memory Test

The first part of the BGP PDU (In) module is the BGP Memory Test. Packets arriving here from the remote peer could represent several functions of the protocol or states of the current peering session. This module does pre-processing based on this information and routes the packets to subsequent modules within the BGP PDU (In) module.

If the remote peering session is up, then all packets are passed. The packet is then checked to see if it is a "Line Down" packet. This would be the case if the remote node's timer expire event that is associated with this node activated. That would mean that this node failed to send the remote node a BGP packet within the BGP Timeout interval. If the packet is a "Line Down" packet, then the local node's memory Line From X Down? is set to yes so that all non-control type outgoing BGP traffic destined to the remote node is sunk. The local node also generates a "Line Down Received" packet to the offended remote node so that it can complete its BGP session shutdown process. If the packet is not a "Line Down" packet, then it is a normal BGP update or keepalive message and it is passed to the BGP In Processor module (see Figure 37).

If the remote session is down, then the only packets that are passed into the BGP Memory Test module are BGP session control packets. These packets can have one of three values: "Line Down", "Line Down Received", and "Line Up".

If the packet is a "Line Down" packet then this means that both BGP peers have timed-out nearly simultaneously which is not likely except where there is an extreme amount of traffic on the network causing large delays and stale data. But to provide processing under this state, the node with the highest Node Number will become the slave and allow the other node to control session dis-establishment, network recomputation, and session re-establishment. If the current node is the slave, then it also sets the memory parameter Stop Proc: Timer Expire (event). This memory parameter controls the execution of the BGP Timer Expire module (see Figure 38).

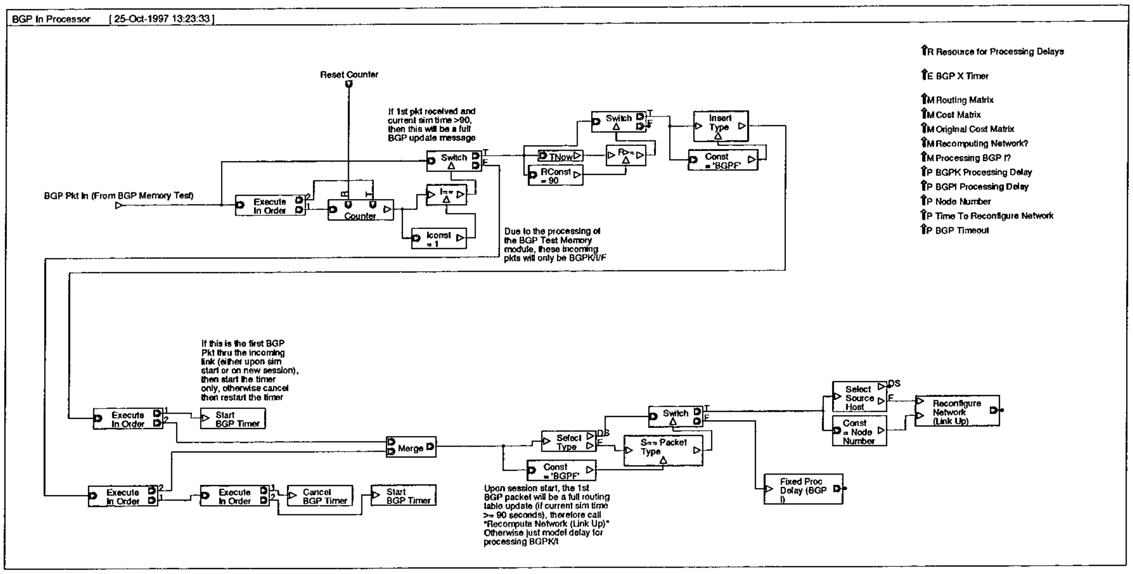
If the packet is a "Line Down Received" packet, then no simultaneous loss of BGP peering session has occurred. In this case, the remote node has offended this node by not sending a BGP packet within the BGP Timeout value. This node's timer expired with respect to the offending remote node and has sent that node a "Line Down" packet. The remote node is now responding with a "Line Down Received" packet. When this packet is received, the local node sets the Line Down Received Packet (Sent From X) memory parameter to yes. This parameter also controls processing that is occurring in this node's BGP Timer Expire module.

If the packet is a "Line Up" packet, then this node was the offending node and is waiting for session control from the remote node. This packet lets the current node know that the BGP peering session is back up and traffic can begin to be sent/received. An important aspect of this branch of execution is that counters within the BGP In Processor module (see Figure 37) are reset, so that the next BGP packet to be received after a failed session is re-established, can be designated as a BGP Full update (BGPF) message. This is in accordance with the protocol behavior [22]. Finally, the memory parameters that have been set as a result of processing a failed BGP peering session are reset to their default values.

Parameters:

- Stop Proc: Timer Expire (memory, non-local): Only used in the case where two BGP nodes have their peering sessions expire simultaneously. In that case, both node's BGP Timer Expire modules are executing. In order for proper session negotiation and network reachability information to be computed, one node has to take over and be master. This memory parameter allows this to happen.
- Line From X Down? (memory, non-local): Controls the execution of this module as explained above. It is also used to sink normal (non-session control) BGP traffic to affected peers.
- Line Down Packet Sent X? (memory, non-local): Used to test for simultaneous session timeout between this node and the remote peer. This memory parameter is set to "Yes" by the BGP Timer Expire module, (see Figure 38), in the event that a timer expires.

- Line Down Received Packet (Sent From X) (memory, non-local): Used as a signal that it is safe to proceed with BGP session shutdown.
- BGP X Timer (event, non-local): Gives this module sight into the timer expire event.
- Timer Expired From Node (parameter, non-local): Used to determine the Node Number of the remote node in the case where both nodes have had nearly simultaneous timer expire events occur and the determination has to be made, based on Node Number, which host will become master.
- Node ID (parameter, non-local): The Node Number of the current node. It is used as explained under Timer Expired From Node above.



If 1st pkt received and current sim time >90, then this will be a full BGP update message

Due to the processing of the BGP Test Memory module, these incoming pkts will only be BGPK/I/F

If this is the first BGP Pkt thru the incoming link (either upon sim start or on new session), then start the timer only, otherwise cancel then restart the timer

Upon session start, the 1st BGP packet will be a full routing table update (if current sim time >= 90 seconds), therefore call "Recompute Network (Link Up)" Otherwise just model delay for processing BGPK/I

Figure 37. BGP In Processor

The BGP Memory Test module passes only BGP update or keepalive messages to this module. If the BGP message arriving here is the first one received after a peering session, which has failed is now active, the packet is designated as a BGPF message and processed accordingly. In order to smooth out start up transient behavior in this model, if the packet is the first one through after simulation initialization and the current simulation time is less than 90 seconds, then it is *not* designated a BGPF message. This allows the model to begin simulation "in the middle" of a live internet session. Note that all nodes receiving a BGP full update message simultaneously is not reflective behavior of the internet as would be the case at simulation startup if this check were not used in the model. That is, the real internet doesn't get turned on every morning before it goes to work. The 90 second time is important because that is the value of the BGP Timeout variable which is equal to the actual timeout value recommended in [22]. Using a longer period would jeopardize the correct operation of the model. Using a shorter period would not allow for the correct representation of the Poisson traffic distribution. The BGPF message is then passed to the Reconfigure Network (Link Up) module. The Reconfigure Network (Link Up) module uses the Original Cost Matrix memory structure to reset the network to the condition it was in before the link failed. Regardless if the first BGP message seen is designated as BGPF or not, it also starts the BGP Timer block for the first time. Subsequent messages first cancel the active timer, then restart it for the next message from a particular source host (peering session).

If the message received in this module is not designated a full update, then it is processed according to its type. BGPK messages do not take any time to process and just serve to reset the BGP Timer blocks. BGPI messages take less processing time than do full updates, but still represent overhead to the router's processor as the new topological information contained in the BGPI message has to be processed according to local policy. Eqs (6) and (7) give the derivation of the BGPI processing delay. All BGP messages are sunk after they undergo the appropriate processing delay.

Parameters:

- Resource for Processing Delays (resource label, non-local): Each processing block that references this resource label associates a request for the processor resource to the appropriate instance of a processor. This argument is localized at the Node level.

- BGP X Timer (event, non-local): The event argument associated with the Start and Cancel BGP Timer Blocks.
- Routing Matrix (memory, non-local): Written to by the Recompute Network (Link Up) module which is triggered by the receipt of a BGPF message.
- Original Cost Matrix (memory, non-local): Used by the Recompute Network (Link Up) module to restore this node's view of the network to what it was before the link failed.
- Recomputing Network? (memory, non-local): This memory parameter is exported to the network layer, (see Figure 30), where it is localized and used to indicate whether packets should be delayed through the router.
- Processing BGP I? (memory, non-local): This memory parameter is exported to the network layer, (see Figure 30), where it is localized and used to indicate whether packets should be delayed through the router.
- BGPK Processing Delay (parameter, non-local): The amount of processor time requested of the server referenced by the Resource for Processing Delays by the BGPK process. In this model, no delay is associated with processing a BGPK message.
- BGPI Processing Delay (parameter, non-local): The amount of processor time requested of the server referenced by the Resource for Processing Delays by the BGPI process. The value of this parameter is specific to the router being modeled by the current node.
- Time to Reconfigure Network (parameter, non-local): The amount of processor time requested of the server referenced by the Resource for Processing Delays by the BGPF process. The value of this parameter is specific to the router being modeled by the current node.
- BGP Timeout (parameter, non-local): The amount of time to elapse before a the current instance of the BGP Timer expires and the Service BGP Timer module executes indicating a timer expire event (see Figure 40).

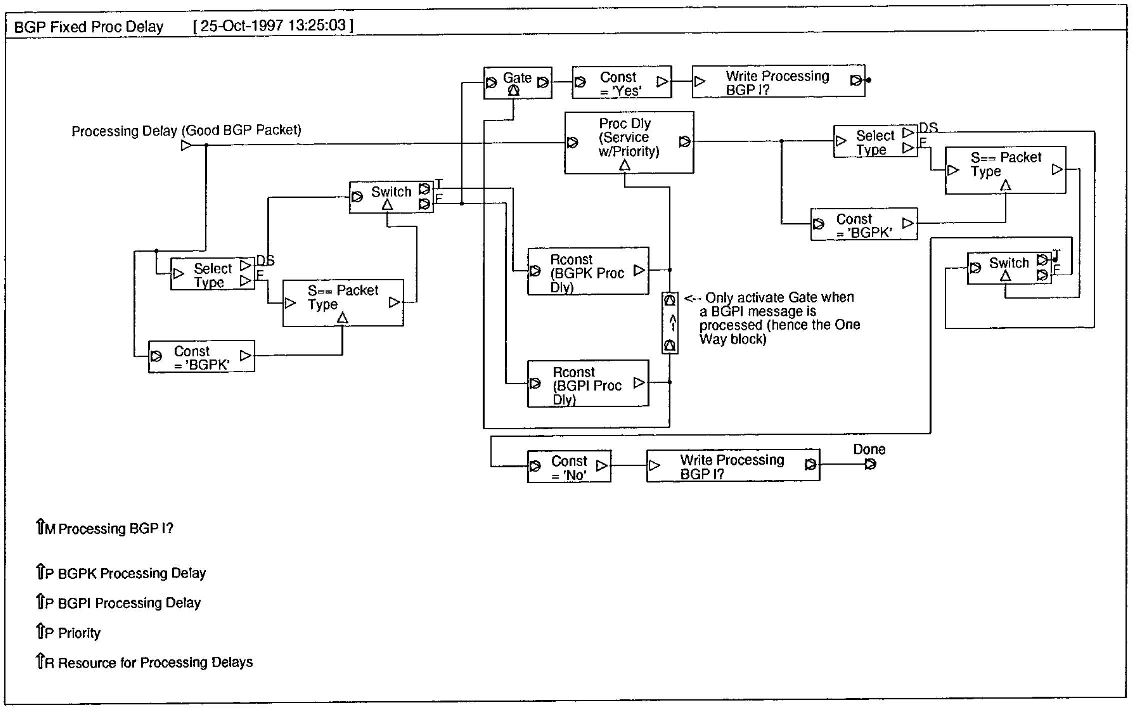


Figure 38. Fixed Processing Delay (BGP)

BGP messages that enter this module are fed into the Processing Delay (Service w/Priority) block. The router's CPU resource is modeled here. Each BGPK/I message that is passed into this block also has service time amounts, (BGPI/K Processing Delay), associated with it. No BGP messages have priority, they are processed in FCFS order. During the time that the BGPI message is being processed the memory parameter Processing BGP I? is set to yes and used at the network layer so that packets traversing the node as a BGPI message is being processed will be delay a fractional amount of time (see Figure 30). Once the BGPI message is finished being processed the memory is reset to no. Note that the Gate block is used to synchronize activating the memory with the beginning of the processing of the BGPI message. Any delay that may be caused by the BGPK message is not modeled in the Transit-Stub ISM.

Parameters:

- Priority (parameter, non-local): Can be set by the user to indicate the priority that is afforded transactions entering the Processing Delay (Service w/Priority) block. All BGP messages are given the same priority in this model.

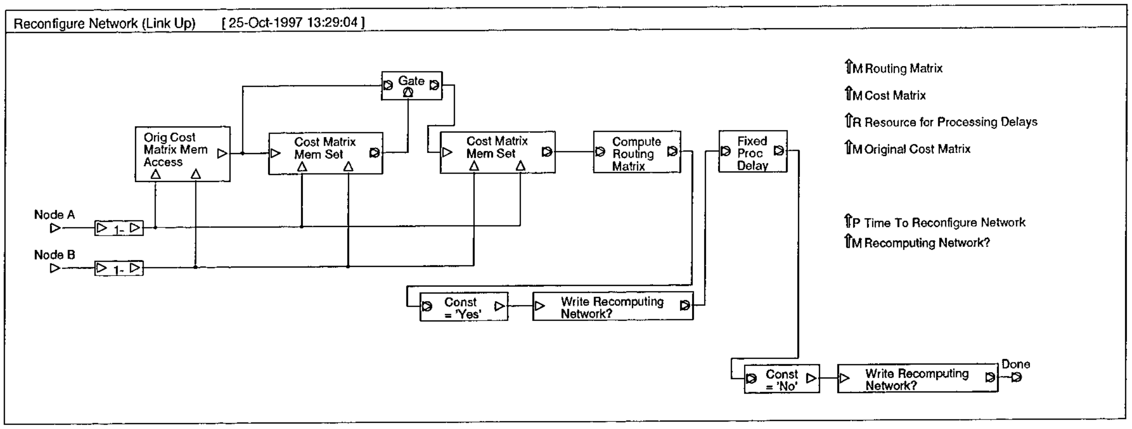


Figure 39. Reconfigure Network (Link Up)

The operation of this module is similar to the previous as far a processing delays are concerned. The same idea is used to set memory parameters that are used at the network layer to schedule packet delay. The Original Cost Matrix memory is used to reset the cost of the failed links, (see the RMatrix Mem Set block), to what they were before the link failed. Note that since the cost matrix is symmetrical (links are bi-directional), the column and row index reference is switched and the Cost Matrix memory is set in the row x column and column x row positions. Note also that matrix data structures in this model are zero-based, that is why the incoming Node Number(s) are decremented. Note also that when simulating the Transit-Stub ISM model, no simulation clock time is used by the Compute Routing Matrix module. The model designer has to specifically account for various processing delays in the model design. This is done with the Fixed Proc(essing) Delay block along with the Time To Reconfigure Network, and the Resource for Processing Delays parameters.

Parameters: Explained before.

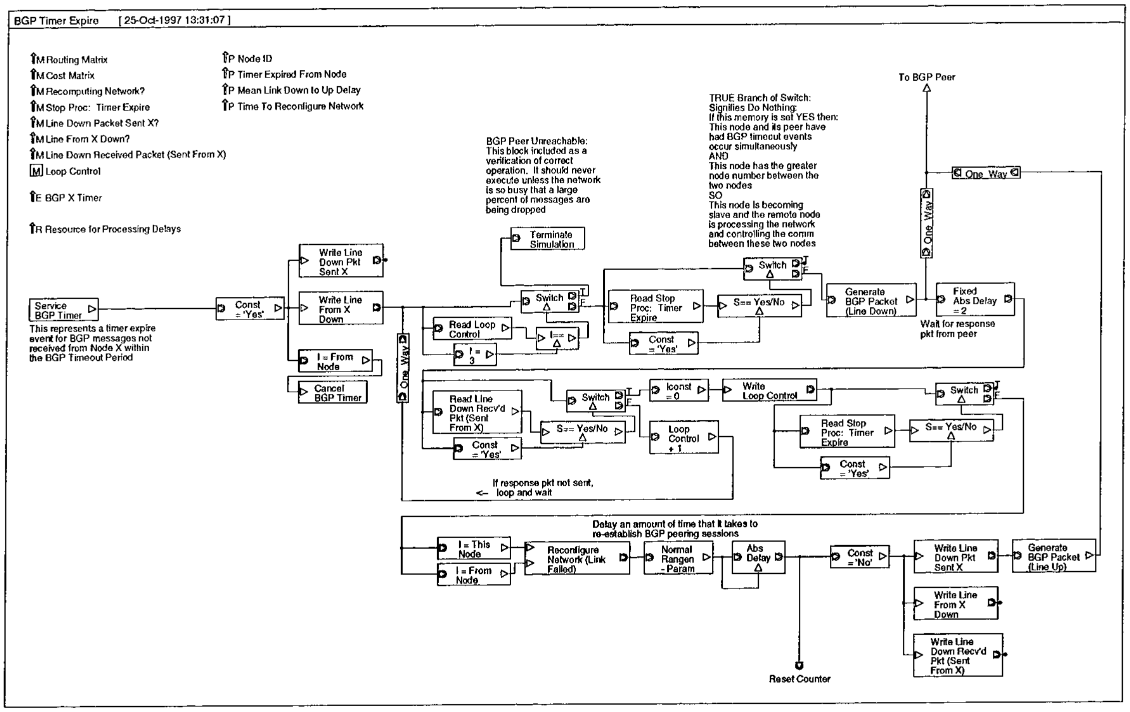

BGP Timer Expire [25-Oct-1997 13:31:07]
M Routing Matrix
M Cost Matrix
M Recomputing Network?
M Stop Proc: Timer Expire
M Line Down Packet Sent X?
M Line From X Down?
M Line Down Received Packet (Sent From X)
M Loop Control
E BGP X Timer
R Resource for Processing Delays
P Node ID
P Timer Expired From Node
P Mean Link Down to Up Delay
P Time To Reconfigure Network
Service BGP Timer
This represents a timer expire event for BGP messages not received from Node X within the BGP Timeout Period
Const = 'Yes'
Write Line Down Pkt Sent X
Write Line From X Down
I = From Node
Cancel BGP Timer
One Way
Read Loop Control
I = 3
I==
BGP Peer Unreachable: This block included as a verification of correct operation. It should never execute unless the network is so busy that a large percent of messages are being dropped
Terminate Simulation
Switch
T
F
Read Stop Proc: Timer Expire
S== Yes/No
Const = 'Yes'
TRUE Branch of Switch: Signifies Do Nothing: If this memory is set YES then: This node and its peer have had BGP timeout events occur simultaneously AND This node has the greater node number between the two nodes SO This node is becoming slave and the remote node is processing the network and controlling the comm between these two nodes
Generate BGP Packet (Line Down)
To BGP Peer
One Way
Fixed Abs Delay = 2
Wait for response pkt from peer
Read Line Down Recv'd Pkt (Sent From X)
Iconst = 0
Write Loop Control
Loop Control + 1
If response pkt not sent, <– loop and wait
Delay an amount of time that it takes to re-establish BGP peering sessions
I = This Node
I = From Node
Reconfigure Network (Link Failed)
Normal Rangen - Param
Abs Delay
Const = 'No'
Write Line Down Pkt Sent X
Generate BGP Packet (Line Up)
Write Line From X Down
Write Line Down Recv'd Pkt (Sent From X)
Reset Counter

Figure 40. BGP Timer Expire

This module is executed only when a BGP Timer module is not reset by the receipt of BGP message within the BGP Timeout period. Within BONeS designer, each activation (instance) of any particular timer module is tracked internally. The Transit-Stub ISM employs 10 timer modules and each is instantiated uniquely with the Node Number of the Source Host of the BGP messages received at the current node. The event argument BGP X Timer is similarly linked. But, given that particular timers and events can be uniquely identified, each timer can be activated many times and each activation has to be tracked individually as does the associated Timer Expire event. That is why this module can be standalone and only has outgoing connections (see Figure 35).

When a timer expires the memories Line From X Down? and Line Down Packet Sent X? are set to yes. The former is used to sink normal BGP traffic between the two nodes and the latter is used in the BGP session negotiation process. The local BGP timer is also canceled in case any

BGP messages “leaked” into the BGP In Processor module after the timer expire event took place (this is unlikely because the interarrival times between BGP messages are large compared to the execution times of these modules).

The processing at this module now enters a loop. If the Stop Proc: Timer Expire memory is not set to yes which indicates that this node and the remote node have timed out nearly simultaneously, (see the discussion accompanying Figure 36), the loop is allowed to continue. A “Line Down” packet is sent to the offending node and this node waits for a “Line Down Received” packet from the remote node. The Fixed Absolute Delay of 2 seconds is ample in this model as representative end-to-end delays are much smaller. If during the delay period, the BGP In Processor receives a “Line Down Received” packet, then the memory Line Down Received Packet (Sent From X) is set to yes, then this module can exit the loop and proceed with its function. If not, it loops, sends another “Line Down” packet and waits for a response. If the loop is not exited after three iterations, the network has become very unstable and the simulation is halted.

When the loop is exited, the loop control variable is reset to zero, the Stop Proc: Timer Expire memory is tested once more for safety and, if it is set to no, processing proceeds. At this point, the BGP session between the two nodes is down. In the real internet, that means that the link can no longer be used to carry data traffic and new reachability information has to be sent out. This is done by the Reconfigure Network (Link Failed) module. After that a delay is encountered that is commensurate with the amount of time that it takes two routers to re-establish a failed BGP peering session. The delay is generated by a random number generator with acting on a normal distribution with a mean equal to the Mean Link Down to Up Delay and a variance of 2 seconds. The Mean Link Down to Up Delay was obtained from [20] while the variance is an estimation on my part. Before the BGP session is reestablished, the counters at both this node and the remote node are reset so that the next BGP message received can be designated as a full update. The memories Line Down Packet Sent X?, Line From X Down?, and Line Down Received Packet (Sent From X)? are reset to no and a "Line Up" packet is sent to the remote node. Upon receiving the "Line Up" packet, the remote node resets the appropriate memories and counters, thus the cycle is ready to repeat.

Parameters:

- Recomputing Network? (memory, non-local): This is a yes/no memory and is used by the WAN BGP Network Layer module to delay data packets when the it is set to yes. The Reconfigure Network (Link Failed) module sets this memory to yes when it is active.
- Stop Proc: Timer Expire (memory, non-local): Used to control BGP session negotiation and network reachability computation in the event that both nodes in the peering session have timer expire events nearly simultaneously. The BGP Memory Test module has the proper scope to see this eventuality and this memory parameter is set there.
- Line Down Packet Sent X? (memory, non-local): The normal operation of this module is to send a "Line Down" packet to the node that caused a timer expire event. When this is done, this module sets this parameter to yes.
- Line Down Received Packet (Sent From X) (memory, non-local): This parameter is set to yes in the BGP Memory Test module when the remote node, responding to a "Line Down" packet from this node, send this node a "Line Down Received" packet in response. This event lets the current node know that it is OK to proceed with reconfiguring network reachability information for this model (to execute the Reconfigure Network (Link Failed)) module.
- Loop Control (memory, local): Used to control the loop that allows this node to wait for a response to the "Line Down" packet.

- Node ID (parameter, non-local): Equal to the Node Number of the current node. It is used as input to the Recompute Network (Link Failed) module.
- Timer Expired From Node (parameter, non-local): Equal to the Node Number of the remote node that caused the timer to expire. It is used as input to the Recompute Network (Link Failed) module.
- Mean Link Down to Up Delay (parameter, non-local): This parameter is supplied at the simulation system level. It is a value, in seconds, used as input to a normal distribution-based random number generator. The value itself was supplied from [20]. The normal distribution has a variance of 2 seconds which is an estimation. This time is representative of BGP session re-acquisition between peers in the internet. When a BGP session fails, the link between the peers is no longer available to data traffic. When the session is re-established, the reachability capability represented by those peers is re-injected, (via a BGP full update message), into the internet.
- Time To Reconfigure Network (parameter, non-local): This parameter is representative of the time that it takes a router to process a BGP Full update message [20]. It is used as input to the Reconfigure Network (Link Failed) module.
- BGP X Timer (event, non-local): This parameter is localized at the BGP PDU level which allows the proper scope for the management of timer expire events in this model.
- Resource for Processing Delays (resource, non-local): This resource argument is localized at the nodal level. It is accessed here by the Reconfigure Network (Link Failed) module.

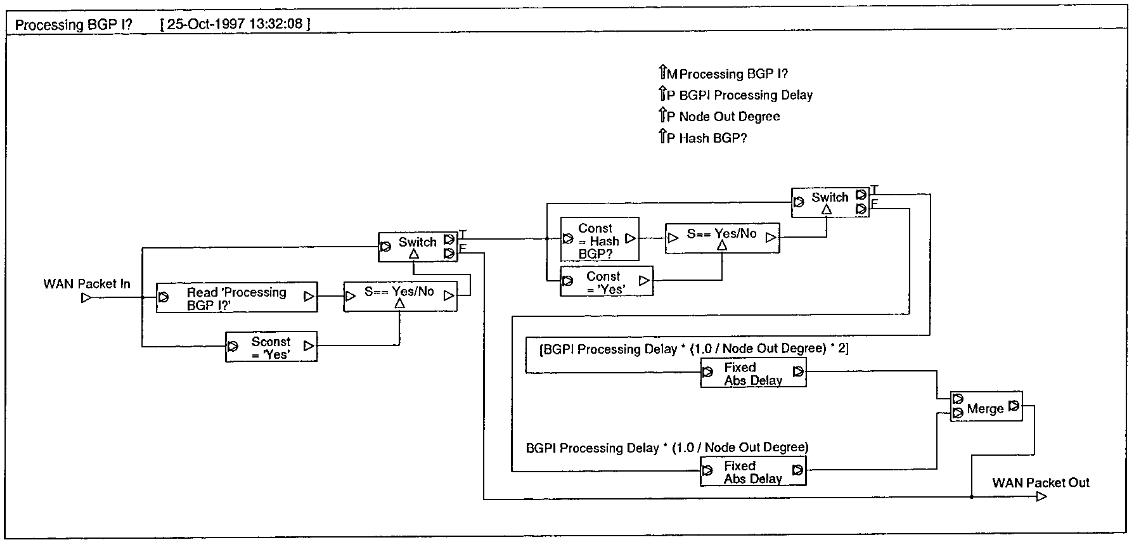


Figure 41. Processing BGP I?

This module is contained within the WAN BGP Network Layer (see Figure 30), and is responsible for delaying data packets while the node is processing a BGPI message. See the discussion accompanying Eqs (7) and (8) for an explanation of the delay times. The data packets are delayed differently depending on the value of the Hash BGP? parameter. If the node is not currently processing a BGPI message whether or not hashing is being used, the data packet undergoes no delay.

Parameters:

- Processing BGP I? (memory, non-local): If set to yes, then data packets are delayed.
- BGPI Processing Delay (parameter, non-local): Set at the simulation system level, this parameter is the base on which the packet delay is computed.
- Node Out Degree (parameter, non-local): The degree of the current node. This parameter is actually mis-named as all the links in the model are bi-directional. It is used in the computation of the packet delay.
- Hash BGP? (parameter, non-local): Two delay values are used, the lesser value is used if the simulation is being run where BGP messages are not being hashed.

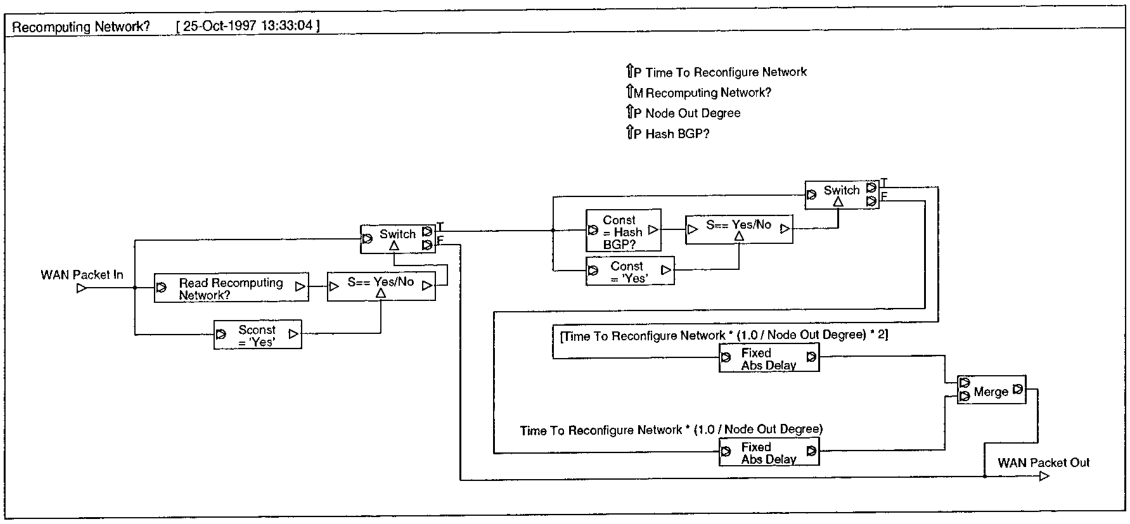

Recomputing Network? [25-Oct-1997 13:33:04]
P Time To Reconfigure Network
M Recomputing Network?
P Node Out Degree
P Hash BGP?
Switch
T
F
Const = Hash BGP?
S== Yes/No
Const = 'Yes'
Switch
T
F
WAN Packet In
Read Recomputing Network?
S== Yes/No
Sconst = 'Yes'
[Time To Reconfigure Network * (1.0 / Node Out Degree) * 2]
Fixed Abs Delay
Merge
Time To Reconfigure Network * (1.0 / Node Out Degree)
Fixed Abs Delay
WAN Packet Out

Figure 42. Recomputing Network?

Similar to the module in Figure 41, this module just uses longer delay values because recomputing full reachability information is more CPU intensive than computing partial reachability information.

Parameters: Explained at Figure 41 and Eqs (7) and (8).

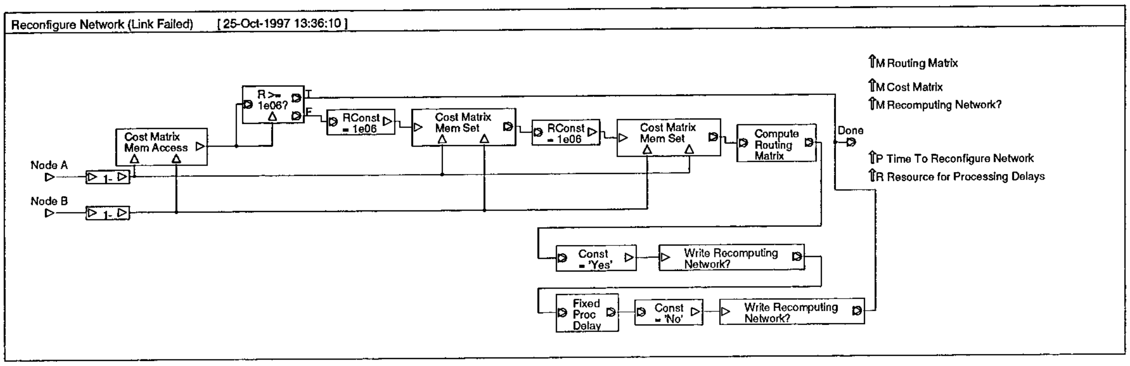


Figure 43. Reconfigure Network (Link Failed)

This module is contained within the BGP Timer Expire Module. The network reachability information is recomputed based on a failed BGP peering session. The link that has failed has its cost set to 1E6, (which represents infinity in this model), by this module. The Compute Routing

Matrix module is then called and executed over the new information. During the time that the network is being recomputed the memory variable Recomputing Network? is set to yes. This allows the nodes to delay data packets appropriately.

Parameters:

- Routing Matrix (memory, non-local): The memory that contains the routing information for the network. It is modified by this module.
- Cost Matrix (memory, non-local): This memory represents link costs for the entire network and is modified by this module prior to being used by the Compute Routing Matrix module.
- Recomputing Network? (memory, non-local): Set to yes for as long as new network reachability information is being computed.
- Time to Reconfigure Network (parameter, non-local): Used as input to the Fixed Proc Delay block to simulate the amount of time it takes to recompute the network. This time value is coupled to the:
- Resource for Processing Delays (resource, non-local): Note that in the BONeS environment, the actual execution of the Compute Routing Matrix module is not modeled as part of the simulation time. It has to be "assigned" an appropriate delay. This delay is modeled by the Fixed Proc Delay block.

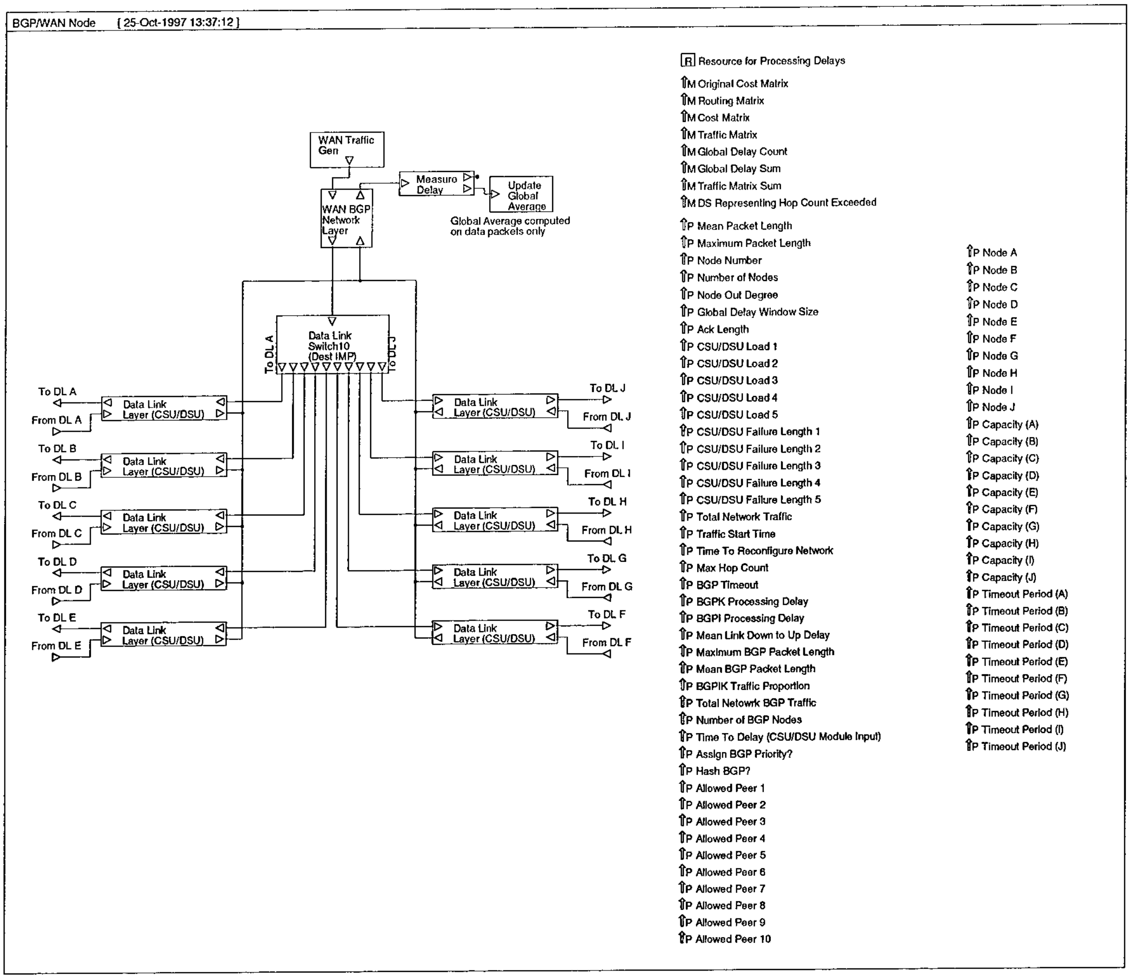


Figure 44. BGP WAN Node

This module and the next are the basic building blocks of the Transit-Stub ISM. These represent router objects in the internet. The resource argument Resource for Processing Delays is localized here so that all processing that its subordinate modules do can be represented on a nodal basis. These modules along with the Full Duplex Link get added to a system-level module that becomes the simulation system (see Figure 46). The WAN Node is also responsible for measuring the delay of all data packets that are delivered to the transport layer from the network layer. The delay statistics are gathered by subtracting the time the packet was created from the current simulation time. These delays are averaged over a window the whose size is given as a parameter in the simulation system (see Figure 46). The number of averages taken during the simulation is also controlled by a parameter of the simulation system. In this model, the window size is purposefully chosen to be small compared to the time between averages so that there would be a smaller "smoothing" effect by the windowed averages and truer delay behavior could be reported in the simulation statistics.

In addition to the Resource for Processing Delays, the following parameters are instantiated when adding this module to a simulation system module to give the modular control over the configuration of the router objects:

- Node Number (parameter, local): The number of the current node in the model.
- Node Out Degree (parameter, local): The number connections to other nodes.
- CSUDSU Load 1 … 5 (parameters, local): Gives thresholds at which point these units start to inject bit error into packets.
- CSU/DSU Failure Length 1 … 5 (parameters, local): The amount of time to delay incoming packets based on the amount of loss being modeled.
- Time To Reconfigure Network (parameter, local): The amount of time that it takes the node to process a full BGP update. These times vary based on the processing power of the router object. The higher in the hierarchy that an node is placed, the bigger it is. Nodes 1 through 4 are the largest in the model and their value for this parameter is 40 seconds [20].
- BGPK Processing Delay (parameter, local): Due to the minimal processing required of this message type, this value is equal to 0.001 throughout the Transit-Stub ISM.
- BGPI Processing Delay (parameter, local): The Time To Reconfigure Network parameter and Node Out Degree parameters are used to compute this time.
- Mean Link Down to Up Delay (parameter, local): The amount of time that it takes the router object to re-establish a failed BGP peering session. The minimum amount of time in this model is 15 seconds [20] with larger values for nodes lower in the hierarchy.

- Allowed Peer 1 ... 10 (parameter, local): Used to control the establishment of BGP peering sessions in the model.
- Node A ... J (parameter, local): These parameters are used to reference the remote node on the outgoing ports A through J. The Data Link Switch (Dest[ination] IMP) uses these values to switch the packets to the correct outgoing ports. In Figure 46 for example, node 1 is connected to node 5 via port D, so the parameter Node D is given the value "5" when it is instantiated for node 1. Conversely, node 5 is connected to node one via port A, so the parameter Node A is given the value "1" when it is instantiated for node 5.
- Capacity (A ... J) (parameter, local): These are the capacities of the various links in bits per second. The Data Link Layer module uses these parameters to model transmission delay. The CSU/DSU behavior module uses these to measure throughput. Several probe modules use the capacity parameter as well.
- Timeout Period (A ... J) (parameter, local): The value used by the timer block in the Data Link Layer module. These timers control processing associated with ACK packets in the model. The timeout value is given by Eq (1).

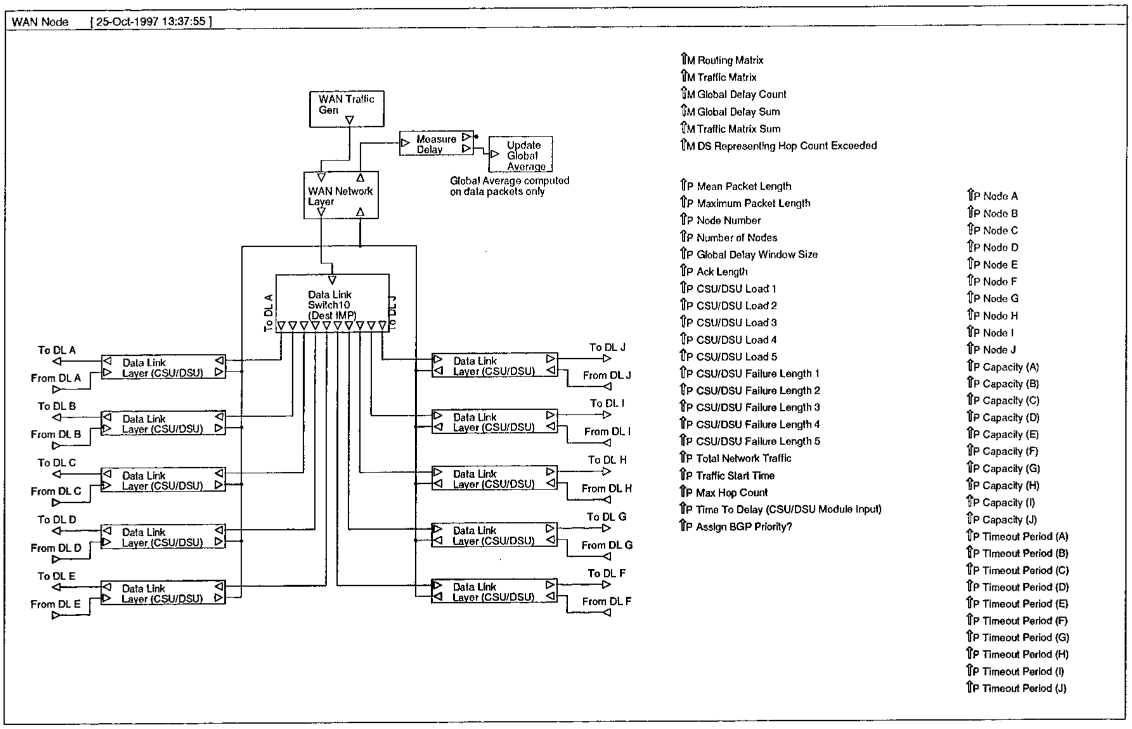


Figure 45. WAN Node

The WAN Node is identical to the BGP WAN Node except that it has no BGP processing capabilities. As such, it is used to represent intra-AS internet operation in this model. The parameters for this module are explained at Figure 44.

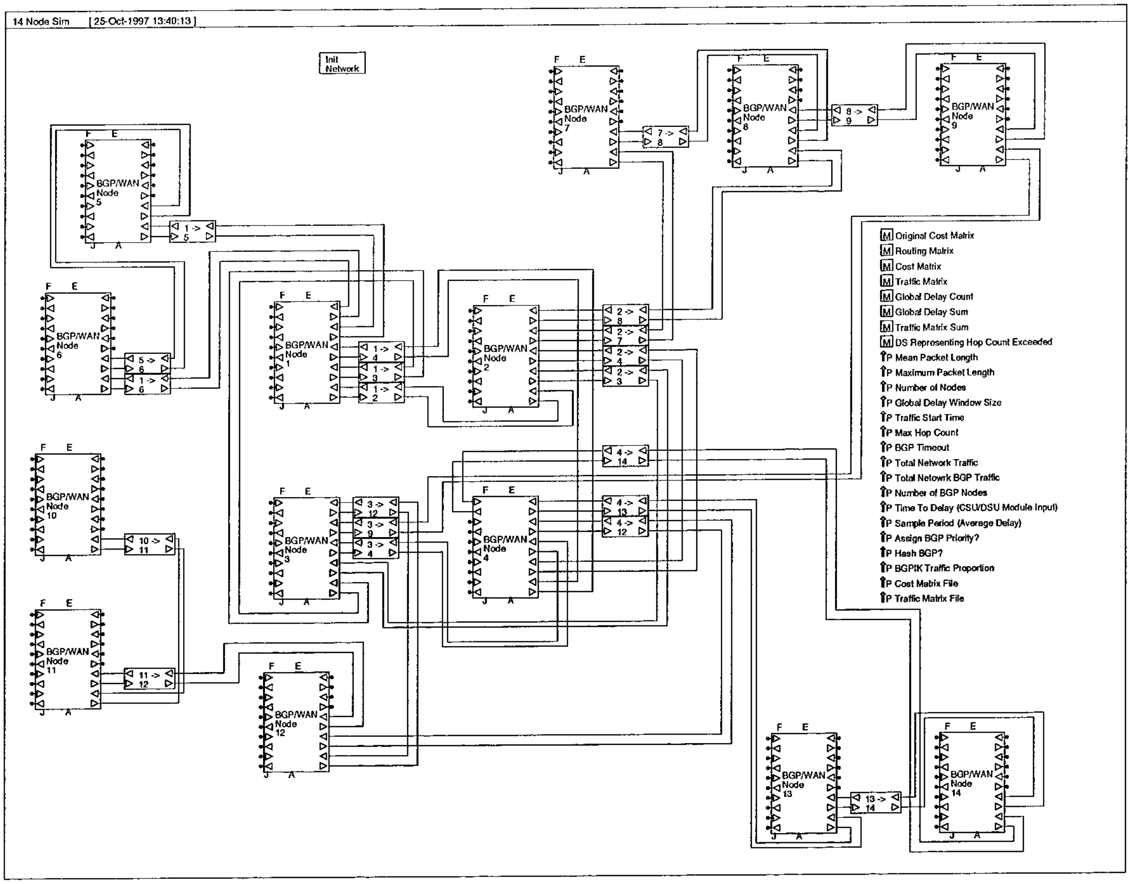


Figure 46. Transit-Stub ISM Simulation System (14-Node)

The simulation system is exercised by BONeS Designer and the results are reported in their Post Processor module based on probes that are inserted into the model at simulation time. This model uses 117 probes placed at various points within the model. The types of probes are:

- Network Delay Probe: this probe is inserted in the R/ block of the Compute Global Average module (see Figure 18). It is a generic probe that simply gathers the output of the module so that the information can be retrieved and plotted with the BONeS Post Processor capability.
- Max Hop Count Exceeded Probe: This probe is place in the DS Representing Hop Count Exceeded Memory (see Figure 44). This memory is written at the network layer if packets exceed the Max Hop Count of the model. A trend in this direction could be

interpreted to mean that the network was having an excessive number of BGP session failures and re-establishments. This would produce a high number of network updates and packets traversing the network during this time may get routed to more intermediate nodes than under normal operating conditions.

- Mean Hop Count Probe: this probe is placed at the "to transport layer" exit port of the network layer in the node model (see Figure 44). Like the Network Delay Probe, it produces a windowed average of the hop counts of all packets that will be retired at the transport layer. Again, the window size is small compared to the frequency of averages taken which allows a truer representation of the data.
- Link Utilization Probe: these probes are placed on the links after the propagation delay has been modeled in the Full Duplex Link module. These probes use the Length field of the WAN packet along with the capacity of the current link to determine utilization. The data is reported as averages over a window period. Again, the window period is small compared to the number of averages reported.
- Link Transmission Queue Statistics: these probes are placed in the transmission queues of the data link layer. BONeS provides a primitive block that outputs the final queue statistics upon simulation completion and these probes are actually placed there. The information being gathered is a snapshot of the queue's operating state over the life of the simulation. Representative statistics available through this probe are: max number in queue, mean delay through queue, mean number in queue, and number of packets rejected.
- Number of Dropped BGP Peering Sessions: The probes are placed at the "to data link layer" exit port of the network layer module. They trap BGP "Line Down" packets that are sent by the BGP PDU when a peering session times out. As data traffic rates are

increased during the simulation runs, this probe will give information showing how the BGP traffic is affected.

- Retransmitted Packets/BGP Packets Received: The probes are placed at the "from data link layer" entrance port to the network layer. It is a generic probe whose filters are set to only copy retransmitted packets (data and BGP), or BGP packets. With this the total number of retransmissions can be shown and since the full data structure is copied, any field of the data packet can be queried to find out the operating characteristics of the model when the packet was retransmitted. The probe also copies all BGP packets on the link. Full BGP session information is available via this information.

All parameters of the simulation that were not accounted for have to be localized (if it is a memory argument), or instantiated (if it is a variable) here. These are the ones that should have vision over the entire simulation system. Included in these parameters are the main control variables in the simulation: traffic rate, the use of hashing on BGP traffic, and the assignment of priority to that traffic. Where parameters are given more than one value means that the simulation will be run in an iterated fashion using those values. The number of iterations for a single global seed will be 16 (4 traffic rate values * 2 hash BGP values * 2 prioritize BGP values).

Parameters:

- Global Delay Count (memory, local): each time a data packet passes into the transport layer a tally is incremented. This tally becomes the denominator supplied to the Compute Global Average (network delay) module.
- Global Delay Sum (memory, local): each time a data packet passes into the transport layer its delay value is added to this memory. This total becomes the numerator supplied to the Compute Global Average (network delay) module.

- Mean Packet Length: (parameter, local): 1120 bits

- Maximum Packet Length: (parameter, local): 12000 bits

- Number of Nodes (parameter, local): 14

- Global Delay Window Size: (parameter, local): 'Sample Period (Average Delay)' * 5.0 simulation seconds. This parameter controls the window size over which the packet delays are averaged.

- Traffic Start Time (parameter, local): 1E-4 simulation seconds

- Max Hop Count (parameter, local): 8. This parameter is set to the ceiling of ('max hop-based diameter' + 'average hop based diameter') which is equal to 7.24.

- BGP Timeout (parameter, local): 90 simulation seconds

- Total Network Traffic (parameter, local): 150, 300, 450, 600 packets/simulation second. This does not count BGP or ACK traffic.

- Number of BGP Nodes (parameter, local): 14

- Time To Delay (CSU/DSU Module Input) (parameter, local): 10 simulation seconds

- Sample Period (Average Delay) (parameter, local): 'TSTOP'/100 simulation seconds. This parameter controls the rate at which global packet delay averages are taken within the window.

- Assign BGP Priority? (parameter, local): Yes/No

- Hash BGP? (parameter, local): Yes/No

- BGPI/K Traffic Proportion (parameter, local): string that supplies the path/filename that contains the information used to feed the cumulative distribution random number generator.

- Cost/Traffic Matrix File (parameters, local): string that supplies the path/filename that contains the information used to load the Cost and Traffic Matrix memories.

In addition to the simulation parameters, each node has a particular instantiation of the parameters governing how it processes BGP traffic. These parameters are reflective of individual peering policies that are meant to mimic a representative policy picture in the internet. The following table lists the attributes.

Table 12. Instantiation of the Nodal BGP Parameters

Node #	TTRN (sec)	KPD (sec)	IPD (sec)	MLD (sec)	NOD
1	40	0.001	40 * (1/21) * 2	15	5
2	40	0.001	40 * (1/21) * 2	15	5
3	40	0.001	40 * (1/21) * 2	15	5
4	40	0.001	40 * (1/21) * 2	15	6
5	80	0.001	40 * (1/21) * 2	30	2
6	80	0.001	40 * (1/21) * 2	30	2
7	80	0.001	40 * (1/21) * 2	30	2
8	60	0.001	60 * (1/21) * 1.2	22.5	3
9	80	0.001	40 * (1/21) * 2	30	2
10	80	0.001	40 * (1/21) * 2	30	1
11	80	0.001	40 * (1/21) * 2	30	2
12	60	0.001	60 * (1/21) * 1.2	22.5	3
13	80	0.001	40 * (1/21) * 2	30	2
14	80	0.001	40 * (1/21) * 2	30	2

TTRN = Time to Reconfigure Network (also equal to BGP Full Update Message Processing Delay)
KPD = BGP Keepalive Message Processing Delay
IPD = BGP Interim Update Message Processing Delay
MLD = Mean Link Down to Up Delay
NOD = Node Out Degree

Appendix B: Model Revisions

The following are my informal notes on the revisions to the model as it was being built. These notes are included here as an aide to anyone doing follow-up research. They will be particularly helpful further modifications are made to the model using the BoNES Designer software. The format of the revision notes contains the revision data, the module that was revised, and an explanation of the revision. The remarks are in order by date.

9/11/97

Data Structure: WAN Packet

Added "Line Down Received", "Line Up", and "Line Up Received" to the status set of the WAN packet. This will be used to control the interaction between two WAN BGP nodes when the timers at any BGP node expire and a loss of session has to be negotiated, the network recomputed, and the session brought back up

9/11/97

Data Structure: WAN Packet

Added The field "Control" to the packet type description in the WAN Packet. This will help parse the packets At the BGP Protocol Module.

9/11/97

Module: Data Link Switch 10 (Source IMP)

This module switches packets from the data link layer in the BGP Protocol Unit. This is so sessions can be managed individually within the "router" (or the network layer). This switch is only employed to route BGP packets. The other switches work on the Destination IMP.

Basically just expanded this module to 10 ports vice 4

9/11/97

Module: Data Link Switch 10 (Dest IMP)

Modified Data Link Switch4 (Dest IMP) ... added 6 more ports

Switches packets to the correct data link (based on the Destination IMP (next hop) which has just been set in the network layer).

Basically just expanded this module to 10 ports vice 4

9/12/97

Module: BGP Timer Expire

Changed the timer handle to Destination host because the data structure being used to control the timer handle is a "Line Down Received" packet. This packet is a reply to the original "Line Down" packet the peer sent as a result of a timer expire event on that end.

9/12/97

Module: BGP Memory Test

Re-wrote BGP memory test. Most of the functionality is explained within the diagram itself. This module is used on incoming BGP traffic is meant to be employed on each separate BGP Connection. It tests for various states that BGP peers could be in with respect to each other. In the case of lost BGP sessions, this module ensures that if two peers time out together, then only one peer will re-compute the network. The Network recomputation is done for {source X destination} and {destination X source} simultaneously because of the global nature of the memory accessed by this model.

Non-global memory availability can still be modeled from the standpoint of instantiating the variable "Time To Recompute Network" on a nodal basis and setting its value to a value which is commensurate to the size of the Autonomous System to which the current node belongs.

Also, even though only one node is recomputing the network (which is an artifact of the model and simulation), traffic delay can be introduced at both nodes as if the routing matrix updates (network recomputations) were taking place at both nodes simultaneously.

9/12/97

Module: Fixed Proc Delay (BGP)

Modified Fixed Proc Delay (BGP) to remove the parsing for the BGP Full Update Packet. The Full update packet causes a re-computation of the network. The Full update packet is received only once during the lifetime of the BGP session, and that is at the beginning.

Previously, the Timer Expired event would recompute the network twice: once on BGP session loss (link down) ... it would then delay an exponential amount of time ... and then recompute the network (link up). It is better to recompute the network in this module because it makes that re-computation a function of the BGP traffic (which is what it should be) and not just some exponential delay which is not as reflective of the situation as this approach is.9/12/97

Modified Fixed Proc Delay (BGP) to remove the parsing for the BGP Full Update Packet. The Full update packet causes a re-computation of the network. The Full update packet is received only once during the lifetime of the BGP session, and that is at the beginning.

Previously, the Timer Expired event would recompute the network twice: once on BGP session loss (link down) ... it would then delay an exponential amount of time ... and then recompute the network (link up). It is better to recompute the network in this module because it makes that re-computation a function of the BGP traffic (which is what it should be) and not just some exponential delay which is not as reflective of the situation as this approach is.

9/13/97

Module: BGP Timer Expire

This module is part of the BGP In Protocol Unit

It executes as a result of the interarrival timer for a BGP message expiring from any connected BGP node. There is one of these units for each connected BGP node. The multiplicity of these units is caused by the BONeS implementation of the Timer Blocks. They have to be tracked on a per peer basis.

This module recomputes the network because of a lost BGP peering session between any two nodes. It then delays an exponentially generated random amount of time (with the mean equal to a reasonable representation of the time it takes for border routers in the internet to re-establish a dropped BGP peering session.

Then it brings the connection back up

In the case of the timer's expiring at two peering neighbors simultaneously, this block will only execute if this node is the master node in the relationship. The master node is the node with the smallest node id. This seems like an arbitrary heuristic, but in this model it is not, because it is instantiated where the nodes with the smallest node IDs are the largest/topmost in the hierarchy border routers.

9/13/97

Module: Generate BGP Packet (Line Down)

Changed module to write Packet Type "control" instead of packet type "BGPK"

9/14/97

Module: BGP Out Processor

This is a modification of the BGP Out module (now contained in the WAN Example>Archaic library). It is more modularized which supports the ease of adding units to support more than 10 simultaneous BGP sessions from/to any neighbor.

--> Brief Explanation of Why modules are duplicated with in this module

Argument for grouping 10 BGP Out Processors in this BGP PDU (Out) module is explained in with the BGP PDU (In) module. Basically, since each BGP peering session has to be tracked independently of the other, and since certain BONeS blocks require separate instantiations, this (or the BGP PDU (In)) module couldn't be compressed any more.

--> Explanation of Reset Counter Input Port

This port received a trigger signal from the BGP PDU (In) module upon the re-establishment of a previously failed BGP session. The counter is reset to allow the first packet out to be the Full BGP Update.

When the remote node receives the full BGP update from any peer, it causes that remote node to recompute the network.

Of course in this, model with its global memory, the full network is recomputed, but this model can also constrain the amount of time it takes to recompute the network to be a time commensurate with the size of the AS to which the remote node is attached. That way, more realistic data traffic patterns can be observed as a result of more realistic network update overhead in the mesh.

9/15/97

Module: Recomputing Network?

This is a queue DS

For every data packet that enters a BGP node at the network layer, the local memory variable Recomputing Network is queried. If the network is currently being recomputed, every packet that enters this module will be queued (initial input is held) and a an integer memory will be updated to reflect the total number of packets entering the queue. These packets are held for as long as the network is being recomputed.

Once the network is done being recomputed, the memory value Recomputing Network is set to false and data packets will take the false branch at the first switch. This allows the memory value Current Q Occupancy to be queried. If that value is greater than or equal to one, then the queue release mechanism is triggered. A packet leaves the queue and decrements the Current Q Occupancy memory. This continues to happen as long as there are packets left to leave the queue. The trigger mechanism is first enabled by a state transition from recomputing network to not recomputing network. Thereafter the action of a packet leaving the queue triggers the queue release mechanism. As long as there are packets left in the queue and the node does not re-enter the recomputing network state while the queue is being emptied, then the queue will continue to be emptied.

If the node returns to the recomputing network state while the queue is being emptied, then all packets are held (including the ones presently in the queue from the last time that the network was being recomputed) until the node re-transitions back to the not recomputing network state. During normal operation, the queue will have ample time to empty before the node can re-enter a recomputing network state.

9/15/97

Module: WAN BGP Network Layer

Added module Recomputing Network?

9/16/97

Module: BGP/WAN Node

Updated this node to have a max degree of 10

The following parameter arguments should be set as this module is added to a system level module:

_ Node Number: Instantiates the number of the node

_ Time To Reconfigure Network: This parameter should be set as a function of the current node size (layer in the hierarchy of the simulation system implies larger capacity, processing power, queue sizes, etc.) and the degree of the current node and the sizes of the nodes attached to this node. This parameter only exists at BGP nodes.

_ Mean Link Down To Up Delay: this set as a function of the size of the current node and number and size of peers. This parameter only exists at BGP nodes.

_ Node (X): Instantiate as the node numbers to which this node is connected

_ Capacity (X): The capacity of the link in bits/second which connects this node and node X

_ Timeout Period (X): set as a function of distance to remote node (propagation delay), max packet size and link capacity of the link connecting remote node X (transmission delay).

9/19/97

Module: Generate BGP WAN Packet

Added functionality to sink a BGP packet if it is destined to a node with a relationship with the host node as described below. I have set up capability for this model to support two such relationships from any one node. That is, from any host BGP node connected to any destination BGP node, at most two of those destination nodes can have the no-peer relationship established. Note that this relationship is symmetrical.

In the case where an AS is multi-homed to a single provider via different border routers in that AS, then the two internal routers do not need to share internal BGP information, so those two BGP nodes within the same AS would not need to set up a BGP peering session. A strictly internal protocol would suffice for keeping reachability information updated.

9/19/97

Module: Fixed Proc Delay (BGP)

Added memory argument "Processing BGP I?". This argument is set while the node is processing a BGP Interim Update message. The processing times for this message is based on the parameter argument "BGPI Processing Delay" (which is set at the nodal level because it can take on different values for different BGP nodes).

The memory argument "Processing BGP I?" is exported to the nodal level. While active (= yes), data packets traversing that node will undergo a delay based on a percentage of the time represented by the parameter BGPI Processing Delay. Note that the packets won't undergo the full delay because a router object can still switch packets while it is updating its IP routing tables. The delay will be modeled by an absolute delay module.

9/19/97

Module: Processing BGP I?

This module will delay packets a fraction of the time that is instantiated in the parameter "BGPI Processing Delay", if at the time a data packet enters this module that the memory argument "Processing BGPI?" is true. Otherwise packets undergo no delay.

This module will reside at the network level and is a part of the WAN BGP Network Layer module.

9/22/97

Module: Fixed Proc Delay (BGP)

Added One-Way block to connection from "Rconst (BGP I Proc Dly)" to the "Processing Delay" blocks. This is to force the correct operation of the gate which enables the memory trigger to fire at the right time. Without the One Way block, the "Rconst (BGPK Proc Dly)" block would operate the gate release port and would allow the memory to be set when other than a BGP I packet is being processed. This is not the intended operation of the model as BGP Keepalive messages are being processed without incurring delay.

9/22/97

Module: CSU/DSU Behavior

This module was modified to only affect traffic in the outgoing direction. This is because this module is deployed on both sides of the data link at the network layer. Note that traffic throughput is still computed on two-way traffic however.

9/26/97

Module: WAN Nodes:

Had to make link capacities of unused data-link layers = 1 (bit per second) instead of zero. This is because of the Uniform Pulse Train in the BONeS-supplied primitive "Throughput". Even though a Data Link Layer might not be used for a particular node (each node has capacity of a degree of 10 even though it may not be attached to that many nodes), the BONeS-supplied

Throughput module still fires in the Data Link Module causing division by zero. Making the link value one will hurt nothing because nothing is ever accessed/output within the unused data link layers.

9/27/97

Module: BGP Timer Expire

Added a One_Way block between the output of the "Generate BGP Packet (Line Down)" block and the "To BGP Peer" port. This is to prevent the output of the "Generate BGP Packet (Line Up)" block from looping back into the upper part of the module.

9/27/97

Module: Fixed Proc Delay (BGP)

Added logic to only write "no" to the "Processing BGPI?" memory by a BGP I packet. Previously, both a BGPK and BGPI packet could write "no" to this memory. Since only a BGPI packet could activate the memory, having a BGPK deactivate it could cause the memory to be deactivated prematurely. This change will ensure that the memory stays activated (set to "yes", for the length of time indicated by the "BGPI Processing Delay" parameter.

9/27/97

Module: BGP In Processor and BGP Out Processor

Added and "Execute In Order" block to remedy the ambiguous race condition occurring at the counter block. See page 3-43 in the Modeling Reference Guide for an explanation.

9/27/97

Module: CSU/DSU Behavior

Changed the "Capacity" parameter of the "Throughput" block from 0.5 * "Capacity" to "Capacity" since the output of the Throughput block is a percentage of overall capacity and the two outputs in the CSU/DSU Behavior block are being added together to get link full duplex throughput.

Also changed the module to route the data structures into an absolute delay module instead of a sink when link utilization crossed certain thresholds (defined by the CSU/DSU Load X parameters). This more closely resembles the end-to-end network packet delay experienced at the session level when packets which have been corrupted by the CSU/DSU units have to be retransmitted. Note the use of the "Memory Switch" Blocks which have an OR input requirement. This will allow the full flow of the data packets because a packet hitting this switch is switched immediately based on the current value of the switch port (bottom port). These memory switches are initially set (on startup) so that the link utilization capacity is assumed to be below the lowest threshold value (CSU/DSU Load 1).

9/27/97

Module: Processing BGP I? & Recomputing Network?

Added parameter "Node Out Degree". This will be used to control the amount of absolute delay packets undergo when the node is processing a BGP I message. It will be:

BGPI Processing Delay * (1/Node Out Degree)

9/28/97

Module: Reconfigure Network (Link Failed)/(Link Up)

Rerouted the output from "Fixed Proc Delay" into the memory un-set block. This was done so that the block didn't terminate before the memory had been written

9/28/97

Module: WAN Packet Tx Delay

Changed the Iconst = Capacity block to Rconst = Capacity Block. Removed the associated int-to-real block.

9/30/97

Module: Data Link Layer

Added an Execute in Order block after the Service Packet Timer block. This was done to help keep the counters between the Start and Canx Packet Timer block synchronized.

9/30/97

Module: Test Sim

Changed the Max Packet Length parameter to 12000 bits (1500 bytes)

9/30/97

Module: Generate BGP WAN Packet

Removed No Peer 1 and No Peer 2 parameters and added Allowed Peer 1/2/3/4/5/6/7/8/9/10 which are used to filter outgoing BGP traffic. This is because the model was incorrect before as it would generate BGP traffic to more than just one-hop BGP neighbors. The traffic generation module was left unchanged, that is, it still generates BGP traffic to all BGP nodes indiscriminately, so this filter will allow only one-hop BGP neighbors to establish a BGP session.

9/30/97

Module: BGP Out Processor and BGP In Processor

Moved the counter logic from the out processor (if the counter was equal to 1 meaning that the packet was the 1st packet generated after simulation start or after a previously failed BGP peering session is brought back up, then the packet type was set to BGP Full Update) as it was not correct there (that is because that packets were being generated to multiple destinations but with the counter in the BGP Out Processor, only the first packet would be a BGP full update ... this is incorrect as the first packet *received* at a node after session start is considered to be a full update) and moved it to the head of the BGP In Processor (where each session is tracked individually by neighbor id).

10/8/97

Module: ACK/Data Switch

Removed logic testing for "Data" packets as it was passing data packets and sinking all BGP traffic!!!!! Now all non-ack packets are going out the "Data Out" port.

ROOKIE MISTAKE!!!!!

10/13/97

Module: BGP In Processor

Replaced the execute in order and counter blocks leading up to the Start and Cancel Packet timer blocks. Now, make decisions based on selection of packet type (which is better knowledge).

10/13/97

Module: Timestamp (Data Link Layer)

The constant value of OK was being inserted into the Status field of BGP control packets as they crossed the link. This is not correct. Changed the logic to insert OK into the Status field of Data packets only!!

10/14/97

Module: BGP Timer Expire

Changed the module controlling the amount of time that the link was down after the Reconfigure Network (Link Failed) module executes.

Previous Module: Exponential Gen (mean: "Mean Link Down to Up Delay")

New Module: Normal Rangen Gen (mean: "Mean Link Down to Up Delay", variance: 2).

10/21/97

Module: Hold Buffer

Placed an execute in order block so the packet in the queue would be released before the switch was tripped. Also added value 'Retrans' to the Status field of packets which are retransmitted.

10/21/97

Module: BGP In Processor

Added logic to produce a BGPF update packet only if the counter = 1 and the current simulation time >= 90 seconds (the value for BGP timeout). This will allow the startup behavior (BGPF updates being sent all at once) to be overcome. Basically, the model can be turned on and entered in the middle of operation.

22 Oct 97

Module: Hold Buffer (and Data Link Layer)

Added logic to not overwrite the Status field of BGP Control packets which are being retransmitted. Added timestamp block to module and rerouted to not go through the Timestamp module in retransmission path (as this module would reset status to OK).

22 Oct 97

Module: 14 Node Sim

Changed link capacities for links 3 - 12 and 11 - 12 to 256000 bits/sec (this is to cure bottleneck)

23 Oct 97

Module: 14 Node Sim

Set BGP traffic to 6 packets/second permanently. This is the minimal amount of traffic to keep sessions active between all nodes in the network.

24 Oct 97

Module: BGP In Processor

Re did the logic to have the first packet thru set the timer whether or not that packet is a BGPF update packet. See the comment at 21 Oct.

24 Oct 97

Module: Packet Priority (DL)

Added parameter Assign BGP Priority?. This parameter is set at the simulation system level and controls the assignment of priority to BGP traffic during the entire simulation. Its default value is no. If set to yes then BGP traffic is transmitted with the same priority as is ACK traffic.

25 Oct 97

Module: Processing BGP I? & Recomputing Network?

Added the parameter argument Hash BGP? (Yes/No data type) and blocks to delay the data packets differently if this is set to yes. This parameter is set at the simulation system level and is constant for the entire simulation.

3 Nov 97

Module: 14 Node Sim System

Change link capacities so that the link utilization percentages would be more in line with the 50-node simulation results. The changes are in an excel file on my computer.

Bibliography

1. Antonov, Vasim. Chief Technical Officer, Pluris, Inc., Palo Alto, CA. Personal Correspondence. 1 October 1997.

2. Applegate, Bob. President, Water Wheel Systems, Malrton, NJ. Personal Correspondence. 22 September 1997.

3. Cain, Ed. Network Engineer, Defense Information Systems Agency Center for Systems Engineering, Washington, D.C. Personal Correspondence. 19 September 1997.

4. Carter, CDR Clarence E. and others. The Man in The Chair: Cornerstone of Global Battlespace Dominance. AF 2025 Research Paper, April 1996.

5. Department of the Air Force. Global Engagement: A Vision for the 21st Century Air Force.

6. Duncan, Lt Col Bruce and others. The Public Switched Network: An Overview and Vulnerability Assessment. Air Command and Staff College Research Paper May 1995 (ACSC/DE/087/95-05).

7. Govindan, Ramesh and Anoop Reddy. "An Analysis of Internet Inter-Domain Topology and Route Stability," Proceedings IEEE Infocom 1997, Kobe, Japan: IEEE Press, 1997.

8. Halabi, Bassam. Internet Routing Architectures. Indianapolis: Cisco Press/New Riders Publishing, 1997.

9. The Internet Society. "Internet Statistics." WWWeb, ftp://ftp.isoc.org/isoc/charts2/growth/ (28 January 1997).

10. Kent, Stephen and others. Internet Routing Infrastructure Security Requirements Analysis. BBN Technologies Report Number 8141. 8 February 1997.

11. Kent, Stephen. Network Engineer, BBN Technologies, Cambridge, MA. Personal Correspondence. 8 August 1997.

12. Kuhn, Richard D. "Sources of Failure in the Public Switched Telephone Network," IEEE Computer Magazine, 31-36 (April 1997).

13. Labovitz, Craig, G. Robert Malan, and Farnam Jahanian, "Internet Routing Instability," Proceedings of The ACM SIGCOMM 1997. ACM Press, 1997.

14. Labovitz, Craig. Network Engineer, Merit Network, Inc., Ann Arbor, MI. Personal Correspondence. 25 August 1997.

15. Libicki, Martin. "Defending Cyberspace and Other Metaphors." WWWeb, http://www.ndu.edu/ndu/inss/actpubs/dcom/ (February 1997).

16. Longstaff, Tom and Howard Lipson. "Toward the Design of Survivable Systems." CERT Coordination Center, WWWeb, http://www.cert.org/research/ch2.html (June 1996).

17. Martin, Cynthia. Network Engineer, Defense Information Systems Agency Center for Systems Engineering, Washington, D.C. Personal Correspondence. 20 May 1997.

18. Mizuki, Howard. Employee, Delphi Corporation, Detroit, MI. Personal Correspondence. 27 December 1996.

19. Network Design Group. BONeS Designer: Modeling Reference Guide. Foster City, CA: Cadence Design Systems, Inc., 1993.

20. Nygard, Joe. System Engineer, Cisco Systems, Inc., Washington, D.C. Personal Correspondence. 22 September 1997.

21. Paxson, Vern. "End-to-End routing Behavior in the Internet," ACM SIGCOMM 25-37 (August 1996).

22. Rekhter, Y., T. J. Watson, and T. Li. "Request for Comments: 1771." WWWeb, http://asso.nis.garr.it/netdoc/rfc/rfc1771.txt (16 March 1995).

23. Saadawi, Tarek N. and others. Fundamentals of Telecommunication Networks. New York: John Wiley & Sons, Inc., 1994.

24. Stallings, William. "IPv6: The New Internet Protocol," IEEE Communications Magazine, 96-108 (July 1996).

25. Wingfield, Nick. "Router Glitch Cuts Net Access." WWWeb, http://www.news.com/News/Item/0,4,10083,00.html (25 April 1997).

26. Zegura, Ellen W., Kenneth L. Calvert, and Samrat Bhattacharjee. "How to Model an Internetwork," Proceedings of IEEE Infocom 96, San Francisco, CA: IEEE Press, 1996.

Vita

Capt Leif S. King was born on 16 March 1958 in Goldsboro, NC. He graduated from Hudson High School, Hudson, WI in May 1977. He was married to the former Susan Marie McCloud of Paynesville, MN on 16 May 1981. He has two daughters, Aimee, 15, and Abigail, 14. He enlisted in the Air Force on 12 July 1982 and completed his undergraduate work earning a Bachelor of Science degree in Computer Systems from McKendree College in Lebanon, IL in May 1992. He Completed Officer Training School and was commissioned on 18 November 1992.

As an officer, Capt King has been assigned to the Computer Systems Division of HQ Air Force Operational Test and Evaluation Center, Kirtland AFB, Albuquerque, NM from May 1993 until May 1996. While there he performed duties with local area network support and customer help desk support and served as the Chief of the Software Integration Branch. In May 1996, he entered the school of Electrical and Computer Engineering, Air Force Institute of Technology.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1997

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
A MODELING AND SIMULATION APPROACH TO CHARACTERIZE NETWORK LAYER INTERNET SURVIVABILITY

5. FUNDING NUMBERS

6. AUTHOR(S)
Leif S. King, Captain USAF

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

Air Force Institute of Technology
2750 P Street
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCS/ENG/97D-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DISA/JEEBC
Mr. Ed Cain
10701 Parkridge Blvd
Reston, VA 20191-4357

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Air Force Core Competency of Information Superiority will be achieved in an age of decreasing AF manpower and corporate expertise. Increased AF reliance on COTS solutions, coupled with nearly ubiquitous points of entry to communication networks, create unique challenges in maintaining the Information Superiority edge. The protection of the internet is part of this equation. The internet supports the daily business traffic of the Air Force. Personnel, finance, and supply data flow through its routers. Controlling an adversary's access to our information systems, either the data, or the hardware and software that control the data and transform it into information, is a key operation of Defensive Information Warfare which is the primary focus in maintaining Information Superiority. This research will attempt to answer the viability of implementing measures designed to ensure the survivability of the internet communications infrastructure against Denial of Service attacks. It will provide planners the information to make decisions based on the cost and benefit tradeoffs associated with such measures. The requirements of system survivability are a superset of those that ensure security. The Air Force will need the cooperation of outside agencies to build survivability into the systems we rely on, but don't necessarily control.

14. SUBJECT TERMS
Systems Survivability, Denial of Service, Border Gateway Protocol, Autonomous System Internet, Network, Routing, Routing Storm, Flapping

15. NUMBER OF PAGES
191

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL