Performance Evaluation of Telephony Routing over IP (TRIP)

by

Matthew C. Schlesener

B.S.E.E. Kansas State University, Manhattan KS Fall 1996

Submitted to the Department of Electrical Engineering and Computer Science and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master's of Science

_____
Dr. Victor Frost: Chairperson

_____
Dr. Joseph Evans

_____
Dr. Gary Minden

_____
Date of Acceptance

Abstract

The purpose of this document is to provide a detailed understanding of a new signaling protocol being developed for use in the Internet or an enterprise Internet Protocol (IP) network. The protocol is Telephony Routing over IP (TRIP). The most basic function of TRIP is to locate the optimum gateway out of a Voice over IP (VoIP) network into the Public Switched Telephone Network (PSTN) [9]. This document will include a background of signaling protocols, including TRIP, a TRIP test plan to evaluate the attributes of TRIP from a carrier perspective, a description of the TRIP simulation model, performance results and conclusions, and next steps.

Acknowledgments

The author wishes to express sincere appreciation to Professor Victor Frost for his direction throughout the Master's program and his assistance in the preparation of the TRIP simulation model and this thesis. In addition, the author would like to thank Cisco Systems, specifically the TRIP marketing and development team, for their support in the initial stages of model development. And finally, the author would like to give special thanks to Executive Management at Sprint, Network Services for providing this exceptional educational opportunity.

Table of Contents

Table of Figures

Performance Evaluation of Telephony Routing over IP (TRIP)

1.0 Executive Summary

The explosion of the Internet over the past decade has changed the world in many ways. Internet users are able to access a vast diversity of information not previously available. For the telecommunications industry, the Internet revolution has forced a shift toward IP (Internet Protocol) based services. IP provides a flexible framework, which can be utilized to support services from simple file transfer and electronic mail to more complex services like Internet-based gaming and Internet telephony.

Voice over IP (VoIP) services have been available since the inception of the Internet but had no quality of service mechanisms. Network traffic and congestion could cause the voice quality to vary from toll grade to satellite quality or worse. As the Internet matured, consumer demand for integrated IP service offerings grew. This demand for integrated services forced telecommunications providers to address VoIP QoS. The solutions vary from over-engineering IP backbones to mitigate IP congestion to transporting IP traffic over ATM which has built in QoS mechanisms to routing VoIP over a fixed number of circuits to the development of protocols providing IP traffic QoS characteristics [1]. The bulk of this document will be centered on the last alternative, a protocol developed to provide QoS to voice service over IP.

Telephony Routing over Internet Protocol (TRIP) is a telephony routing protocol being developed to provide an IP network with next hop routing information for call requests. TRIP is designed to operate independently of the signaling protocol. This allows network designers the opportunity to implement TRIP in varied network environments. Session Initiation Protocol (SIP) will be the underlying signaling protocol discussed throughout this thesis.

In a SIP network all reachable routes must be manually provisioned in the proxy and gateway. In a medium to large-scale implementation, the manual provisioning of the same routing information twice (gateway + proxy) would be costly and possibly prohibitive. Additionally, the proxy has no knowledge of gateway dynamic state. The lack of dynamic resource information could cause added call blocking to the SIP network. In a TRIP-enabled network both these issues are answered by TRIP-lite, which is an added client application implemented on TRIP-lite enabled gateways. TRIP-lite is responsible for updating the proxy with reachable routes and dynamic resource information. Another area TRIP improves a SIP implementation is location of next hop routing information. SIP uses DNS queries to route SIP requests. TRIP dynamically uses a reliable flooding process to build consistent proxy routing tables. Each TRIP-enabled proxy uses the routing table and is able to locate the optimum path for session instantiation.

The focus of this document is to evaluate TRIP performance through simulation. A TRIP model was developed to assess a TRIP-enabled network while varying physical characteristics of the system such as traffic load and propagation delay (i.e., network

topologies). The performance results were used to draw conclusions about a TRIP network under specific network topology conditions. The evaluation provided results in terms of call blocking probabilities, call request delivery time, and percentage of call request reroutes between TRIP network entities. The TRIP results were compared to SIP performance results. Additionally, the TRIP network was investigated under failure conditions, which provided an understanding of how a TRIP-enabled network will react to the loss of network resources.

The results of the investigation provided several important insights on the performance of TRIP. The conclusions can be used to assist network designers implementing a TRIP-enabled network. The conclusions of this work are listed below.

?? Propagation delay does not impact system blocking probability. In a TRIP-enabled network, the system blocking will be driven by traffic load.
    - This result impacts geographic deployment of location servers to support the network. From a system blocking standpoint, designers do not need to be concerned with propagation delay but must be concerned with traffic load.
?? Overall system blocking will follow Erlang B given the standard traffic assumptions.
    - This result allows designers to implement a correctly sized TRIP network based on forecasted customer usage. This would impact number of trunks to support a given destination prefix, number of gateways in a geographic area, and number location servers in the network.
?? As Location Server-to-gateway (LS-to-GW) delay is increased towards a satellite link delay (250ms), loss of knowledge about the current state of the system causes call blocking to increase at the GW. A carrier will prefer all call blocking to occur at the LS and not at the GW. The reason being that if a call is blocked at the LS, there may be opportunity for the call request to be rerouted to an alternate LS and successfully terminated.
    - This result places a limit on implementation options. TRIP messaging can incur propagation delay equivalent to cross country fiber links but satellite links should not be considered.
?? Propagation delay through network topology, LS-to-GW and Location Server to Location Server (LS-to-LS), does not impact the percentage of reroutes in the system. The traffic intensity is the driving factor.
    - This result dictates that designers be concerned with traffic load and not propagation delay through network topology when addressing TRIP rerouting functionality.
?? LS-to-GW propagation delay will add directly to the call delivery delay. For LS-to-LS delay only a percentage of the propagation delay will add into the total call delivery delay. And that amount will be dependent upon the interarrival rate. As the interarrival rate increases, the TRIP system will be forced to reroute a higher percentage of calls between location servers, which will incur propagation delay introduced between the location servers.
    - This issue impacts the delay budget and network topology. The result indicates that any delay between the LS and GW must be added to overall call setup

delay. While, only a percentage of the delay between LS and LS should be added. And that the delay addition is dependent upon rerouting and traffic load.

?? SIP blocking is consistently higher than TRIP and higher than what would be predicted by Erlang B. This shows that a TRIP-enabled network can achieve better performance compared to a SIP network.
    - This is a very important result in that TRIP provides a SIP network with lower blocking. It benefits the carrier with less provisioning, gateway dynamic resource information available at the proxy, optimum path routing, and also better blocking performance.

?? The time required for a TRIP system to react to a change in state (i.e., gateway trunk failure) is based on traffic load. As the traffic load is increased, the system reaction time to the state change will decrease. Additionally, the results show that propagation delay during a failure scenario does not impact the system reaction to new state.
    - Network failures occur. This result shows that when a failure happens the TRIP network will react within a reasonable time interval and tend toward the new steady state.

This evaluation proved that TRIP is a viable voice telephony protocol and provides benefits over a SIP only network. A carrier implementing a SIP network and planning to offer a voice service should seriously consider implementation of TRIP.

2.0 Introduction

The purpose of this document is to provide a detailed understanding of a new signaling protocol being developed for use in the Internet or an enterprise Internet Protocol (IP) network. The protocol is Telephony Routing over IP (TRIP). The most basic function of TRIP is to locate the optimum gateway out of a Voice over IP (VoIP) network into the Public Switched Telephone Network (PSTN) [9]. This document will include a background of signaling protocols, including TRIP, a TRIP test plan to evaluate the attributes of TRIP from a carrier perspective, a description of the TRIP simulation model, performance results and conclusions, and next steps.

The investigation will center on the impact of varying physical characteristics of the system such as traffic load and network topologies via changes in propagation delay on a TRIP-enabled SIP/IP network. A simulation model was developed and used to evaluate the performance of TRIP. The performance results are used to draw conclusions about a TRIP network under specific load and delay conditions. The model will provide results in terms of call blocking probabilities, call request delivery time, and percentage of call request reroutes between TRIP network entities. Those TRIP results will then be compared to SIP simulation results. Additionally, the TRIP network will be investigated under failure conditions. The results from this line of simulation will provide an understanding of how a TRIP network will react to loss of network resources. The ultimate goal of this thesis and the experiments is to provide an understanding control signaling, specifically TRIP, and to understand how a TRIP-enabled network will react under varying conditions.

The next section will provide a detailed background on several signaling protocols being used today to support varied telephony and data services. The protocols described will range from the predominate PSTN signaling protocol, SS7, to TRIP itself.

3.0 Background

This chapter will provide background information on several control signaling protocols. The objective is to provide an understanding of the evolution of control signaling and set a basis for understanding the functionality delivered by Telephony Routing over IP (TRIP). The protocols discussed will begin with the most utilized control signaling protocol, Signaling System Number 7 (SS7) and then progress into control signaling protocols developed for IP networks. The final protocol described will be TRIP.

3.1 Basic Control Signaling

Control signaling is defined as the system that enables a network to exchange messages related to call setup, monitoring, teardown, and network management information. Control signaling provides the command and control infrastructure for communications networks. It is responsible for coordinating network functions. Early in the evolution of voice communications, signaling traditionally consisted of supervisory functionality (busy status, on-hook or off-hook), addressing (called number), and providing call information (dial tone and busy signals). These control messages had certain characteristics. The characteristics included in-band signaling (i.e., the control signals were transmitted along the same channel as the speech traffic), very long call setup delay (10-20 seconds), and limited call control information. The advent of electronic processing allowed designers to evolve telecommunications and lessen the impact of weaknesses caused by those characteristics. The introduction of Common Channel Interoffice Signaling (CCIS) by AT&T in 1976 began the modern era of signaling in the Public Switched Telephone Network (PSTN). The signaling system based on CCIS was referred to as CCS6 [2].

CCS6 provided considerable improvement over its processors but it still had significant drawbacks. These drawbacks included limited message lengths and low speed signaling links. The CCS6 limitations lead to the development of Signaling System Number 7 [2]. A detailed description of SS7 is provided in the following section.

3.2 Signaling System Number 7 (SS7)

3.2.1 SS7

SS7 is the network control signaling protocol utilized by the Integrated Services Digital Network (ISDN) services framework. ISDN control information for call handling and network management is carried by SS7. SS7 is a large and complex network designed to provide low latency and to have redundancy in many network elements. The SS7 control-signaling network consists of signaling points, signaling links and signaling transfer points. Signaling links or SS7 links interconnect signaling points. Signaling points (SSP) use signaling to transmit and receive control information. A signaling point that has the ability to transfer signaling messages from one link to another at level 3 (SS7 level 3 will be described in detail later) is a Single Transfer Point (STP). There is a

fourth entity, the Service Control Point (SCP), which acts as a database for the SS7 network. The STP queries the SCP to locate the destination of the calls. The design of the SS7 protocol is such that it is independent of the underlying message transport network. The design of the signaling network is very important in that it will directly impact the availability of the overall system. In general, the network will be designed to provide redundancy for signaling links and for STPs. Figure 1 shows a basic SS7 network.

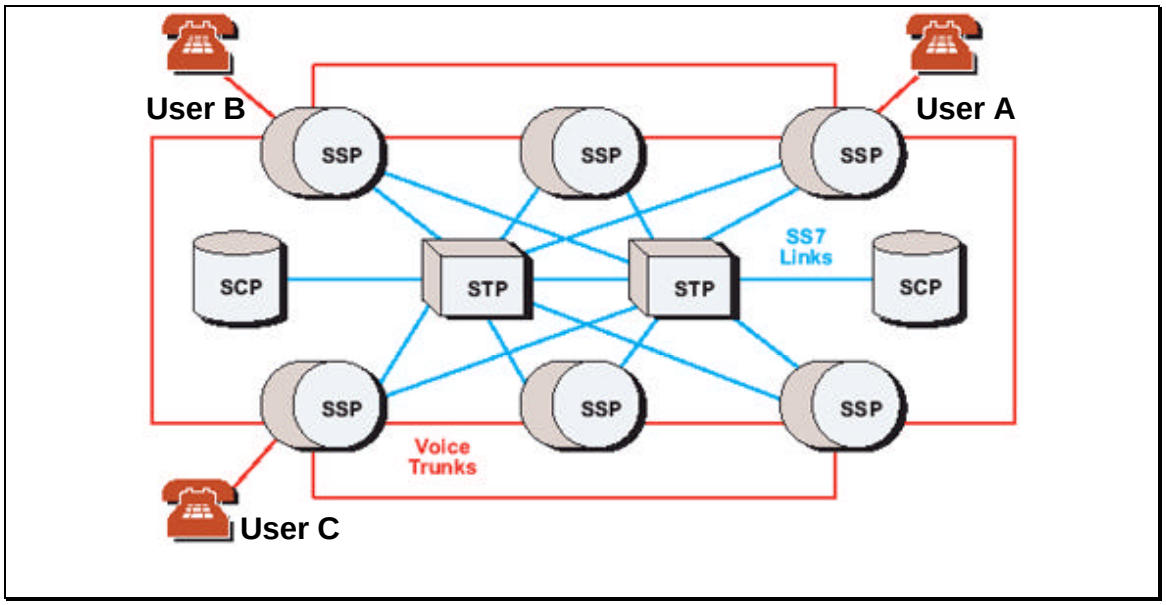

Figure 1: SS7 Signaling Endpoints in a Switched-Circuit Network [3]

A typical call can be illustrated using Figure 1. User A goes off-hook in New York and begins dialing. User A is calling User C in San Francisco. The dialed digits are transmitted across the local loop connection to a local switch that has signal point functionality (SSP). The local switch translates the digits and determines the call is not local to itself. The local switch will use its signal point functionality to signal into the SS7 network to a Signal Transfer Point (STP). The STP queries a SCP to locate the destination local switch. The STP signals to the destination local switch to alert it of the incoming call. The destination local switch rings the phone of User C. User C answers and the two local switches signal across the SS7 network and determine the bearer path through the PSTN. Once the path is setup the call begins. When either user goes on hook, the network signals the other end to tear down the bearer path and the call is terminated. The worldwide SS7 network is divided into national and international levels. This allows the numbering plans and administration to be separated.

3.2.2 Layered Architecture of SS7

SS7 is based on layered protocol architecture. As shown in Figure 2, the structure of SS7 is subdivided into four functional layers. The lower three layers form the Message Transfer Part (MTP). The fourth level is responsible for varied services (e.g., TCAP or ISDP-UP based services) [4].

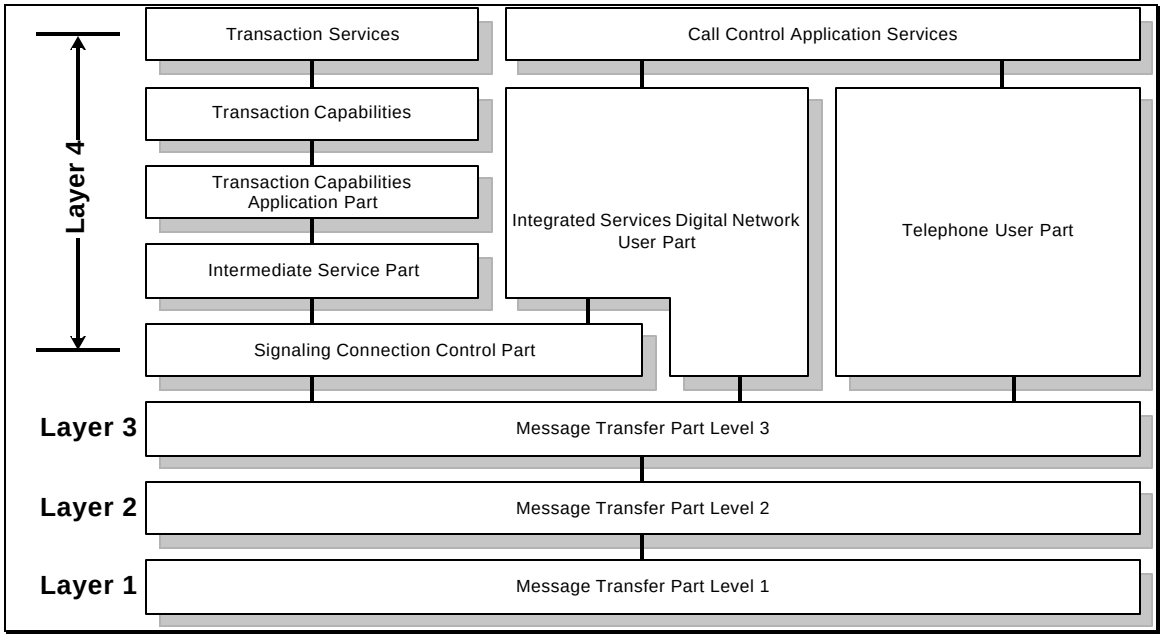

Figure 2: SS7 Protocol Architecture [4]

The MTP corresponds to the first three layers of the OSI model: physical, data link, and network. The three layers of the MTP are called the signaling data link, the signaling link, and the signaling network functions. Together the three layers of the MTP provide connectionless message transfer that allows control information to be transmitted across the network to the destination node. The MTP has the ability to react and take necessary action in response to system and network failure [2].

MTP level 1 corresponds to layer 1 of the OSI model (physical). This level provides the SS7 network with the physical medium to transport control information to the destination node. The signaling data links at MTP level 1 are bi-directional. They consist of two data channels that operate in opposite directions at the same rate. For digital signaling data links the ANSI standard bit rate is 56kb/s [2].

MTP level 2 corresponds to layer 2 of the OSI model (data link). Together with MTP level 1, the signaling link functions provide a reliable link for signaling messages between directly connected signaling points. The signaling messages are transmitted in variable length messages called signal units. Three separate types of signal units exist and the length of the indicator field contained in each signal unit differentiates them.

The MTP level 2 link functions are much like typical data network bit oriented link protocols (e.g., HDLC). The major difference from typical data network protocols is MTP level 2 links are used for signaling. This forces higher performance requirements on the MTP level 2 links. Lost messages, excessive delays, and out of sequence delivery can not be tolerated in a signaling network. Thus, MTP level 2 links must be able to respond quickly to system and network failure. The standard flag (01111110) is used to open and close MTP level 2 signal units and a 16-bit cyclic redundancy check (CRC)

checksum is utilized for error detection. MTP Level 2 signal units employ Fill-in-Signal Units (FISU) when there is no message traffic. FISU are sent instead of sending flags, as done in data link protocols. This provides consistent error monitoring and allows faulty links to be detected even during low traffic load.

MTP level 2 has two forms of error correction. They are Basic Method and the Preventative Cyclic Retransmission (PCR) method. Each method detects errors in signal units, link status message units and FISU. Although, only signal units and link status message units are corrected. Both methods are designed to avoid out-of-sequence and duplicated messages during error correction. In systems with large propagation delay as in satellite networks, PCR is employed. The basic method is usually employed in all other scenarios.

The Basic Method uses the "go-back-N" technique for retransmission. Thus, if a negative acknowledgement is received the transmitter rolls back to signal unit received in error and retransmits everything from that signal unit forward. The sequence numbers for the basic method are seven bits long. Thus, the window size for the basic method is 127 messages.

The Preventative Cyclic Retransmission (PCR) method employs forward error correction. The transmitter holds a copy of each signal unit until a positive acknowledgement is received. When no new signal units are in queue, all signal units not positively acknowledged are retransmitted. Thus, PCR has the ability to accept out of sequence signal units.

The level 2 design incorporates two types of error rate monitoring. The two types are utilized in different phases of signal link usage. The signal unit error rate monitor is used while the signaling link is in service. The signal unit error rate monitor has the criteria for taking a signaling link out of service for excessive error rate. The second type, the alignment error rate monitor, is used when a new signaling link is being brought up. When the link is in the proving state of initial alignment, the alignment error rate monitor has the criteria for rejecting the new link due to excessive error rate [2].

MTP level 3 corresponds to the lower half of layer 3 of the OSI model. They provide functions and procedures for transfer of messages between signaling points. The functions performed at MTP level 3 are divided into two basic categories. They are signaling message handling and signaling network management.

Signaling message handling consists of message routing, discrimination, and distribution. Each signaling point in the network performs these functions. The functions are based on the routing label of each signal unit. The routing label is used to decide what action is to be taken after receipt of each signal unit. The routing label includes a Destination Point Code (DPC) and Originating Point Code (OPC).

Signaling network management functions are to support system recovery during signaling link or signaling point failure. Also, to control traffic when the network is

congested or blocked. The intent is that reconfiguration can occur without message loss, duplication, or put out of sequence. When a change in status of a signaling link, route, or signaling point occurs in the network three functions are activated to reconfigure the system. The functions are signaling traffic management, signaling route management, and signaling link management.

The role of the signaling traffic management function is to divert signaling traffic from unavailable signaling links or routes to alternate available signaling links or routes. The signaling traffic management function also is responsible for reducing traffic in the event of congestion. The role of the signaling route management function is to distribute system information of the signaling network to block or unblock routes. The role of the signaling link management function is to restore failed signaling links, activate new links, and turn down signaling links [2].

The SCCP along with the MTP level 3 provide the functions described by layer 3 of the OSI model. The SCCP augments the MTP addressing with Subsystem Numbers (SSN). The SSN is used by the SCCP to identify each of the SCCP users at a SS7 device. The SCCP also provides an addressing scheme that has global titles. Also, the SCCP provides four classes of service, two are connectionless and two are connection oriented.

The MTP and the Signaling Connection Control Part (SCCP) make up the Network Services Part (NTP). Services can be designed to run across NTP or directly over MTP. The advantage of this design is that only services requiring the support of the SCCP incur higher overhead. Otherwise, services are run directly over the MTP [2].

The SS7 protocol has three major SS7 user parts. The user parts use the transport services provided by MTP and SCCP. The three user parts are the Integrated Services Digital Network User Part (ISDN-UP), the Transaction Capabilities Application Part (TCAP), and Operations, Maintenance, and Administration Part (OMAP).

ISDN-UP provides the signaling functions required to support basic bearer services. These bearer services can be divided into switched voice and data applications. ISDN-UP also supports advanced ISDN and Intelligent Network (IN) services.

The basic bearer service is provided across an access link to the end customer. The user access trunk is logically divided into one signaling channel, the D-channel, and bearer channels, B-channels. The signaling is transported across the access trunk D-channel that employs a separate signaling protocol, Q.931 [5]. The control information provides each signaling point information for setting up and tearing down calls using the access trunk B-channels. Additionally, many ISDN-UP messages have been developed to support service and maintenance during all phases of a call.

The ISDN-UP also provides supplementary services. They include calling line identification (caller id) and call forwarding. Also, provided is user-to-user signaling which is used to support signaling between two user endpoints through the carrier's signaling network [2]. An example is a tie line between two telephone systems at

geographically separated sites of one customer. The signaling channel would transmit signaling information between the two telephone systems and allow them to operate together.

TCAP is a framework of tools in a connectionless environment. The tools are used by one signaling node to execute procedures on another signaling node. TCAP services run over SCCP and MTP. A primary use of TCAP currently is execution of remote procedures in support of 800 services. TCAP functionality will allow the network to learn how to route the call and perform required tasks during each phase of an 800 call [2]. OMAP is provides a SS7 network with protocols and procedures for monitoring, coordination, and control of network resources [2].

3.2.3 SS7 Performance Requirements and SS7 Drawbacks

The performance of a SS7 network is split into three areas. They are availability, dependability, and delay. The availability of a signal route is based upon the components that make up the route and the overall network structure. The dependability of the network is based upon reliable transport of messages. For example, the MTP has a set of objectives for appropriate operation. They are:

- ?? No more than 1 in 10^{10} of all signal unit errors should be undetected.
- ?? No more than 1 in 10^{7} messages to be lost as a result of MTP failure.
- ?? No more than 1 in 10^{10} messages to be delivered out of sequence or duplicated.
- ?? The signal link error rate will not exceed 10^{-6}.

The delay objective is very important to a SS7 network in that delay to signaling information will cause system unsynchronization [2].

SS7 deployed into the PSTN has no major drawbacks or disadvantages. SS7 in the PSTN sets the standard for signaling performance, functionality and reliability. The next section will describe a protocol that was developed to apply the functionality and performance of SS7 to a Voice over Internet Protocol (VoIP) network.

3.3 Voice over IP (VoIP)

3.3.1 VoIP

Voice over IP (VoIP) uses the Internet Protocol (IP) to transmit voice as packets over an IP network. The VoIP service can be offered over any data network that supports IP traffic, like the Internet, enterprise IP networks, and Local Area Networks (LAN). The voice signal is digitized, compressed and converted to IP packets and then transmitted over the IP network. Signaling protocols are used to set up and tear down calls, carry information required to locate users and negotiate capabilities. The main motivations for Internet telephony are very low cost, demand for multimedia communication, and integration of voice and data networks.

3.3.2 VoIP Issues

For VoIP to become widespread, some key issues need to be resolved. Some of these issues stem from the fact that IP was designed for transporting data while some issues have arisen from vendors not conforming to standards. IP was designed to carry data so it does not provide real time guarantees but only provides best effort service. For voice communications over IP to become acceptable to the users, quality of service functionality must be introduced. This can be accomplished through specialized signaling protocols or possibly packet prioritization. Products from different vendors need to operate with each other if voice over IP is to become common among users. To achieve interoperability, standards are being devised and the most attractive options are SIP and H.323. The security problem exists because in the Internet, anyone can capture the packets meant for someone else. Use of encryption and tunneling can provide some security. PSTN and IP telephony networks must be interoperable and appear as a single network. An edge media conversion gateway can perform this task.

3.4 Signal Transport (SigTran)

3.4.1 SigTran

This section will detail Signal Transport (SigTran). SigTran was developed to allow VoIP networks to utilize the extensive functionality and superior performance of SS7. Additionally, a protocol of this nature would allow the telecom industry to reuse the embedded SS7 investment in new revenue generating areas.

The basic architecture for interworking a VoIP network with an SS7 signaling network includes three logical entities. They are the Media Gateway (MG), Signaling Gateway (SG) and the Media Gateway Controller (MGC) [6].

The MG terminates the media streams from the PSTN (e.g., switched voice). It encapsulates the media stream into packets and delivers the packetized voice traffic into the VoIP network. The VoIP network subsequently routes and forwards the traffic to the appropriate host.

The SG is a signaling agent at the edge of the VoIP network that receives and transmits SS7 into the PSTN. The SG has the ability to relay, translate or terminate the SS7 traffic it receives. The SG encapsulates the SS7 signaling into packets and transmits the packetized signaling into the VoIP network using Signal Transport [6]. The signaling packet would generally be destined for a media gateway controller. In many cases, a single physical device provides both MG and SG functionality and a Primary Rate ISDN (PRI) line is used as the connection to the PSTN.

The MGC is responsible for the registration and management of MG resources (e.g., trunks). The MGC is responsible for making session routing decisions based on local policy (e.g., does the user have permissions to use the given service such as long distance). As stated before the MGC is generally the destination of the signaling packets

from the SG. The MGC de-encapsulates the signaling packets and makes call routing decisions. The MGC is configured with IP address to E-164 phone address pairs which allows mapping between the two networks. It locates the destination address (IP or phone number) from the SS7 signaling unit (could be ISDN or Q.931) and signals the IP address of the destination host to the MG. The MG then sets up the bearer path from the host in the PSTN to the host in the VoIP network. The MG notifies the MGC when the call ends. A basic architecture of a SigTran network is shown in Figure 3.

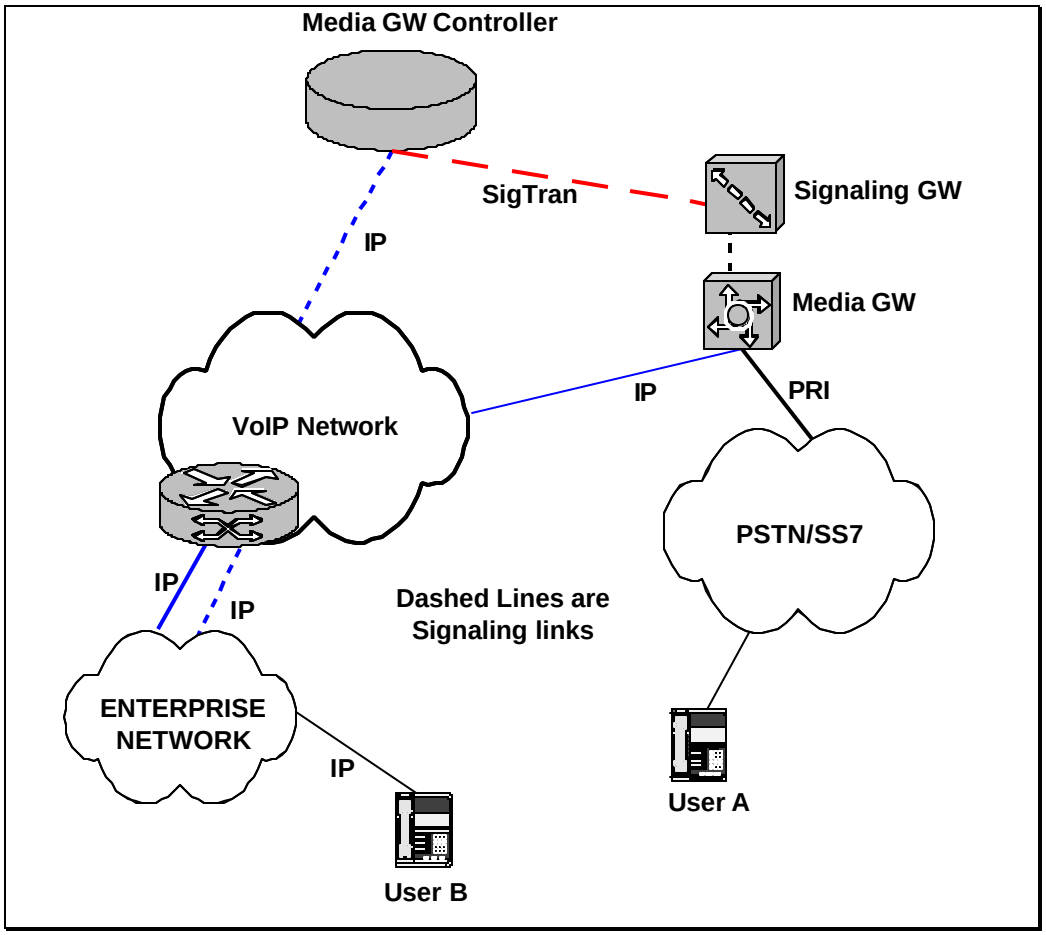

Figure 3: Basic SigTran Network

A basic call using SigTran can be illustrated using Figure 3. User A is connected to the PSTN and is supported by SS7. User B is connected to an enterprise VoIP network. User A wished to call User B. User A goes off-hook and begins dialing User B's E-164 number. The local switch collects the digits and determines the call is not local to itself and sends a SS7 message to a STP. The STP does a database query and determines the call is destined for the media gateway connecting the VoIP network and the PSTN. The STP signals to the media GW across the PRI trunk's D-channel. The D-channel signaling passes through the media GW to the signaling GW. The signaling GW encapsulates the SS7 signaling into packets and transmits the signaling packet to the media gateway controller using SigTran. The MGC has the database where the E-164 number is mapped to an IP address. The MGC locates the IP address and sends the

information to the SG and MG. The SG and MG combine to set up the call though the MG.

3.4.2 SigTran Protocol Requirements

The SigTran protocol is able to support transport for SCCP, TCAP, Q.931, and MTP3. SigTran, like SS7, must ensure in sequence delivery. This is accomplished by using TCP/IP [6].

The IP network must be designed to support low delay and high reliability. Without an efficient network underlying the protocol, the signaling packets will be delayed or lost. As stated earlier the SigTran packets are transported by TCP but the network must have low delay and high reliability to lessen call setup delay. The less reliable the network, the more signaling packets will be lost or out of sequence and that forces retransmissions and higher call setup delay.

The SigTran protocol has been designed to allow flexibility in message length. Depending upon signaling message length, this flexibility may obviate the need for some signaling packet segmentation and reassembly.

Since the underlying network being employed is an IP network, security will be an issue. The protocol was developed to interwork through proxy servers and firewalls. Also, SigTran must interwork with IP Security services.

3.4.3 SigTran Performance Objectives

A SigTran signaling network will be required to adhere to certain performance objectives or undesirable signaling and call behavior will result. The required SigTran network performance for transport of SS7 MTP 3 network management messages is shown below:

?? Message Delay: MTP Level 3 peer-to-peer procedures require response within 500 to 1200 ms. This value includes round trip time and processing at the remote end. Failure to meet this limitation will result in the initiation of error procedures for specific timers.

The required SigTran network performance for transport of SS7 MTP 3 is shown below:

?? Message Loss: No more than 1 in 10^7 messages will be lost due to transport failure.
?? Sequence Error: No more than 1 in 10^{10} messages will be delivered out-of-sequence (including duplicated messages) due to transport failure.
?? Message Errors: No more than 1 in 10^{10} messages will contain an error that is undetected by the transport protocol (requirement is 10^9 for ANSI specifications).

- ?? Availability: Availability of any signaling route set is 99.9998% or better, (i.e.,, downtime 10 min/year or less). A signaling route set is the complete set of allowed signaling paths from a given signaling point towards a specific destination.
- ?? Message length (payload accepted from SS7 user parts): 272 bytes for narrowband SS7, 4091 bytes for broadband SS7.

The required SigTran network performance for transport of SS7 ISDN User Part messages is shown below:

- ?? ISUP Message Delay - Protocol Timer Requirements: one example of ISUP timer requirements is the Continuity Test procedure, which requires that a tone generated at the sending end be returned from the receiving end within 2 seconds of sending an Initial Address Message (IAM) indicating continuity test. This implies that one-way signaling message transport, plus accompanying nodal functions need to be accomplished within 2 seconds.
- ?? ISUP Message Delay - End-to-End Requirements: The requirement for end-to-end call setup delay in ISUP is that an end-to-end response message be received within 20-30 seconds of the sending of the IAM. Note: while this is the protocol guard timer value, users will generally expect faster response time.
- ?? TCAP Requirements - Delay Requirements: TCAP does not itself define a set of delay requirements.

The required SigTran network performance for transport of Q.931 messages is shown below:

- ?? Q.931 Message Delay: Round-trip delay should not exceed 4 seconds. A Timer of this length is used for a number of procedures [6].

3.4.4 SigTran Drawbacks

Signal Transport functions well for its intended application transport of SS7 signaling over an IP network. The major disadvantage to deploying SigTran is that it does not provide a complete solution for signaling in a VoIP network. The direction of SigTran is to provide signaling from a media gateway/signaling gateway to a media gateway controller. The MGC must also be able to control the VoIP end user devices (e.g., IP phones) with a more functional protocol than IP. The next section will discuss a completely different approach to control signaling across an IP network.

3.5 Resource ReSerVation Protocol (RSVP)

3.5.1 RSVP

The signaling protocols discussed to now have been developed to provide network control of end user devices. When an end user wished to utilize a service (e.g.,

make a phone call) the network would provide the control required to set up, supervise and tear down the call as required by the user. The next protocol to be discussed was designed to provide integrated services across the Internet. Resource ReSerVation Protocol (RSVP) was developed to provide receiver-initiated setup of resource reservations for multicast and unicast data flows across an internetwork [8].

3.5.2 RSVP and Quality of Service (QoS)

The Internet was not designed to provide quality of service for applications. RSVP was designed to provide quality of service for distributed real-time applications such as audio and videoconferencing [7]. This is accomplished through signaling from the host into the network. Based on particular application needs, the host will request service with very specific connection parameters from the network. The network routers along the specified path will each be requested for dedicated resources (e.g., bandwidth, etc.). If the router can dedicate the requested resources it will forward the request to the next hop and dedicate the resources for use. If all nodes along the path dedicate the resources, the reservation is complete and the host can begin use. At conclusion, the host will signal that the reservation can be discarded. RSVP ensures QoS along the path by ensuring that each router dedicates all necessary recourses before the connection is setup. If the resources exist, they are reserved and if they are not available, the connection will not be allowed [8].

Additionally, RSVP includes QoS mechanisms to provide traffic control. The traffic control mechanisms include a packet classifier, admission control, and a packet scheduler. The packet classifier is responsible for determining the required QoS class. The scheduler ensures that the guaranteed QoS is delivered. Admission control is one of two modules, policy control, being the other that determine if a reservation is to be set up once requested. The admission control simply decides is the requested resources are available on the local node. Policy control is responsible for seeing if the requesting host owns the required administrative permissions to make the requested resource reservation. If either admission control or policy control checks are returned as failed, the RSVP program returns an error to the host and the reservation is declined [8].

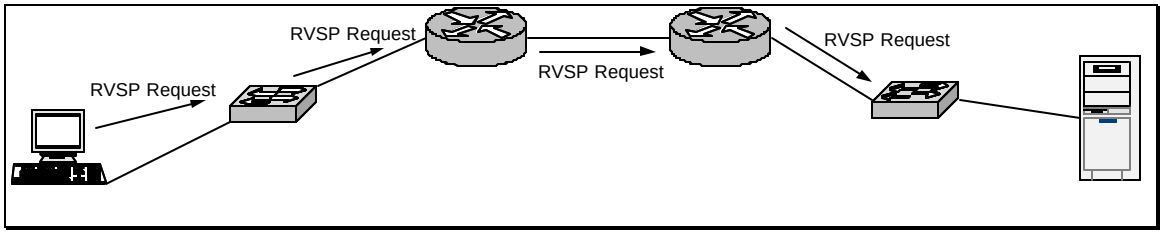

Figure 4: RSVP Resource Request

A simple RSVP call is illustrated in Figure 4, the resource request is sent by the originating host through each hop up to the destination host. If each node along the path makes the reservation, the originating host will commence using the path. If at any node, the QoS mechanisms fail, the path will not be made and the reservation will be declined.

3.5.3 RSVP Messages

The paths requested by RSVP are one direction only. The sending and receiving functions are separate, although they both may be running on the same host at once. RSVP functions above IP. RSVP is not responsible for transport of application information. It only transports control information. This is like Internet Control Message Protocol (ICMP) messaging. Like other management protocols, RSVP will execute in the background and not directly in data forwarding.

RSVP was not designed to be a routing protocol. Thus, RSVP will not aid in route table construction. RSVP will operate with unicast and multicast protocols to set up the path for use by the application layer. If the host wishes to join a multicast videoconference, the host will request the join by sending an IGMP message to group. The host will then send RSVP messaging to reserve resources along the path from the multicast host. Routing protocols have responsibility for forwarding the packets, RSVP will ensure the packets are treated with the appropriate QoS along that routed path.

The RSVP protocol was designed with the dynamic nature of the Internet in mind. The state of RSVP sessions are designed to be built and destroyed incrementally in routers and hosts. Thus, the protocol institutes soft state. Soft state is periodic refresh messaging sent by the host to ensure that all reservations along the path are kept alive. Additionally, this allows RSVP to have timers on the nodes running RSVP. Each reservation has an associated timer. If the timer is not reset by soft state messaging, the reservation will be deleted [8].

3.5.4 RSVP Drawbacks

The major disadvantage of using RSVP is a lack of scalability. The protocol was originally proposed for use in the integrated services packet networks architecture. The aim of which was a network to support both real-time and non-real-time applications. In the article Performance Analysis of an RSVP-Capable Router, based on research the author provides this simple quote “RSVP does not scale enough.” The realization was that RSVP had too much overhead to scale appropriately. The maintenance of state on a per-connection basis proved to be the major scaling obstacle [7]. The next section will describe another network aimed at providing control signaling in a VoIP network. This next protocol begins to look at the entire VoIP network and attempts to provide a complete solution.

3.6 Session Initiation Protocol (SIP) over IP

3.6.1 SIP

As described in then earlier, a SigTran network will suffer from lack of signaling control inside the VoIP network. This is where Session Initiation Protocol (SIP) enters the discussion. SIP is a text-based protocol that resides at the session layer of the OSI model. SIP begins, changes and terminates network sessions [9]. SIP provides advanced

signaling and control to an IP network. SIP supports varied multimedia applications. SIP is designed to efficiently and scalably (unlike RSVP) find network resources based on location-independent name or address and subsequently negotiate session parameters. Along with providing Internet based telephony, SIP is capable of supporting many new services like instant messaging, Internet gaming, and many more.

Within a SIP network, four logical entities exist. They are the user agents, registrars, proxy servers and redirect servers. The user agents are the end users of the SIP network and initiate requests and are the destination of services initiated by other users (e.g., be the called party on a videoconference). IP telephones and PC soft phones (e.g., application software run on a PC that provides telephone like services) are examples of user agents. The registrars are responsible for keeping track of user agents assigned to their network domain. The proxy servers forward SIP requests and responses. The redirect servers take SIP requests and return location information of another user agent or server. In many cases the registrar, proxy and redirect servers are all implemented in the same device. A typical SIP session would involve a user agent initiating a session request through one or more proxy/redirect servers and arrive at the destination user agent [9]. A generic SIP network is shown in Figure 5.

A basic SIP call from the PSTN into the SIP network can be illustrated using Figure 5. User A is connected to a local switch in the PSTN/SS7 network. The SIP Phone User Agent is a registered user at the proxy. User A goes off-hook and dials the SIP phone's E-164 number. The call is signaled through the SS7 network. The SS7 network determines the call is destined for the MG/SG on the edge of the SIP/VoIP network. It signals the MG/SG across the PRI trunk's D-channel. The MG/SG signals the proxy using SIP messages. The MG/SG informs the proxy that a call is being setup to a user agent somewhere in the IP network. The proxy identifies the user agent as part of the enterprise IP network and locates the SIP phone's IP address. The proxy signals the MG/SG with the IP address of the SIP phone and also signals the SIP phone inviting it to start a session and identifies the MG/SG as the terminating end. The SIP phone and MG/SG finish the SIP session. The MG/SG simultaneously sets up the PSTN link back to the local switch where User C is terminated. Once each is complete, the session continues until one user goes on hook and the entire link is torn down.

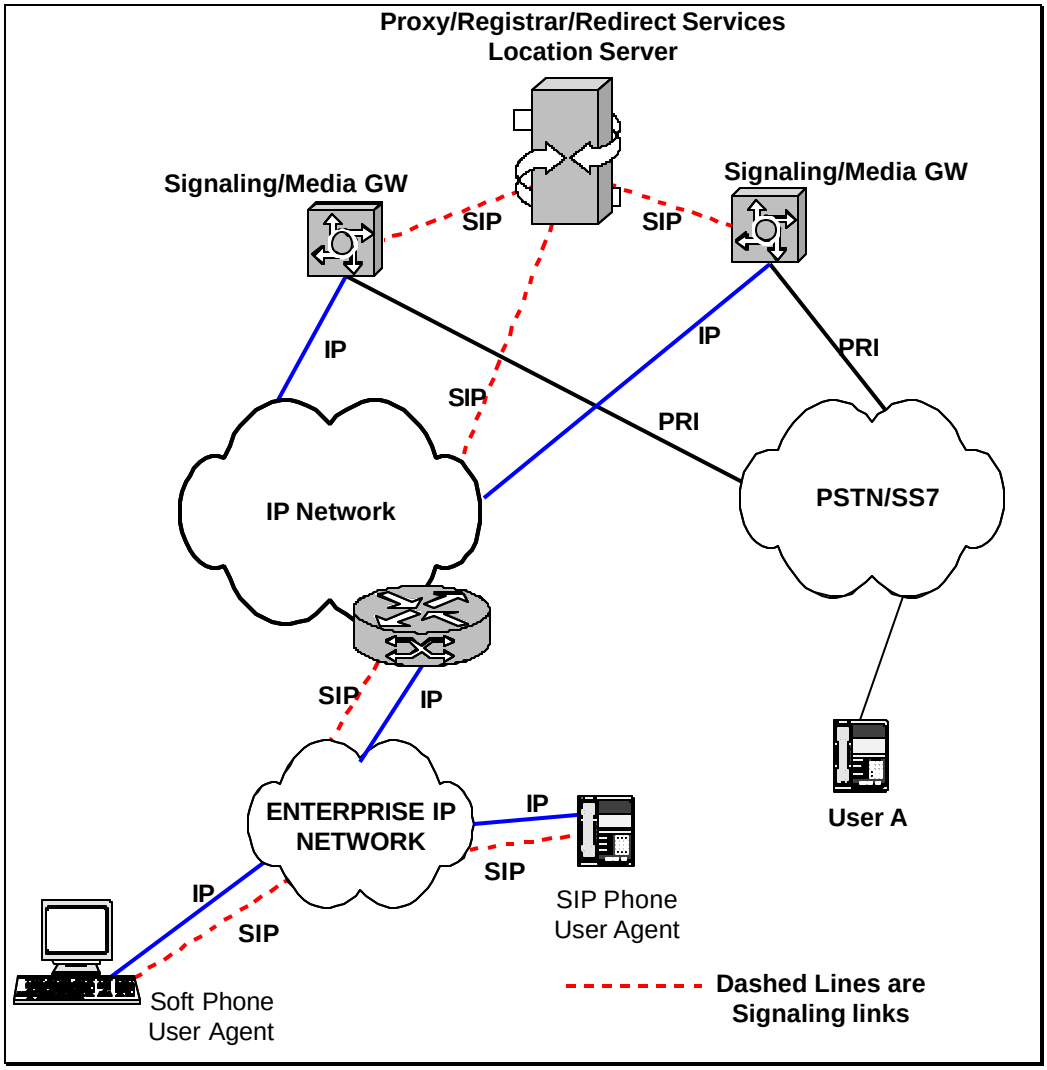

Figure 5: Generic SIP Network

3.6.2 SIP Signaling

The principle role of SIP is to establish sessions between two or more internetwork end systems. The session is then utilized by the end systems to exchange media data driven by the particular application. In general practice at least one of the end systems will be part of an IP domain but this is not required. This implementation would certainly not be a standard implementation [9].

In a SIP network requests are routed using the Uniform Resource Identifier (URI). SIP request URIs look similar to an email address. They include a user and host part as well as a number of parameters. In practice an end user could use their personal email address as their SIP URI [11].

Once a SIP request is forwarded into a SIP network, the proxy and redirect servers will take action based on the URI. If the proxy or redirect servers do take action

based upon a lookup, the URI is rewritten to reflect the new routing information provided by the proxy or redirect server. An example of this process would be if a user placed a forward on his/her SIP phone. Any call request for that SIP phone forwarded to the associated proxy server would rewrite the request URI redirecting the call request to the newly specified destination device [9].

SIP messaging can be transported on a variety of transport mechanisms. The standard implementation would be transport across connectionless User Defined Protocol (UDP). In general this implementation is preferred to circumvent the session setup and tear down overhead incurred with the connection oriented Transport Control Protocol (TCP). Since UDP is generally used the SIP protocol is not supported by any reliable transport mechanisms. To ensure delivery the SIP protocol simply compels the sending host to continually send the specific SIP message until it receives an acknowledgement.

SIP messaging is text based and in general very simple to follow. An invite (INVITE) command is the initial message sent by the call originator. It in essence is inviting the called party to enter the session. The invite will be sent from the originating user agent to a proxy or redirect server. The proxy will subsequently forward the invite request based on its routing table. The destination user agent will send an acknowledgement (ACK) to accept and begin media exchange. Either user agent will tender the bye (BYE) command to terminate the call.

Current SIP implementations utilize Session Description Protocol (SDP) to support multimedia sessions. SDP allows each user agent to declare the type of media streams it wishes to accept and send. Like SIP, SDP is a simple textual format. A typical SDP message will be carried in the SIP message body. Each media stream will include the destination address and port number and a list of received supported encoding schemes.

As with other IP networks, security will be an issue. Authentication is an important part of SIP security. SIP provides basic password authentication as well as digest authentication. Digest authentication utilizes the challenge and response approach that forces the request originator to show knowledge of a shared secret. SIP requests can be signed and verified using Pretty Good Privacy (PGP) [10], which is an application devoted to secure messaging. PGP would be used to verify the identity of the sender but its use would impose substantial overhead on the SIP messaging. As with all IP communications, only end-to-end encryption can provide high confidence of confidentiality.

Mobility is an important area of interest in the communication industry. SIP allows mobility by the ability to locate multiple end addresses for any specified user. If the user defines multiple user agents with multiple addresses the network can terminate sessions at them all [9].

3.6.3 SIP Interworking to the PSTN

Any carrier wishing to provide service from an IP environment will be required to interwork with the PSTN. SIP was designed to provide gateways between the SIP network and the PSTN. As can be seen in Figure 5, the media gateway receives a PRI from the PSTN providing both SS7 signaling and bearer channels. The media gateway does the protocol conversion from SS7 to SIP [9].

In a SIP network, the proxy and media gateways are each manually configured with reachable routes. For example, a gateway has trunks that support a specific destination phone prefix (913-xxx-xxx). That prefix and trunk group information will need to be configured in the gateway. The gateway will then route all calls through that trunk group to that prefix destination. The same prefix information will be manually configured in the proxy also. The proxy will use this information when it receives a call request from either the VoIP network or PSTN. It will look at its routing table and discover it has a gateway with a trunk to the specified destination prefix. The proxy will signal the gateway of the incoming call request and also signal the call originator with the IP address of the gateway. The gateway will then act as the conduit between the SIP VoIP network and PSTN.

Dynamic resources (e.g., trunk group capacity and utilization) on the gateway are not signaled to the proxy. Thus, the proxy has no knowledge of trunk utilization or trunk/gateway failure. It is possible for a proxy to forward a call request to a gateway and have that gateway block the call due to full trunk utilization or failure.

3.6.4 SIP Location Server

A SIP network also includes a logical entity called a location server. The location server (LS) is responsible for locating the next-hop for an incoming session request. The location server co-resides with the proxy and redirect services as shown in Figure 5. For a basic SIP network the LS will use location mappings installed though user agent registration. Each user agent must periodically register its current network address with a SIP registrar service. The registering process allows the LS to know all user agents and associated addresses within its local domain. If the user agent destination is outside the local domain, a DNS look-up is done to locate the next hop information. The use of DNS is a slow process and part of the impetus to define a dynamic routing protocol for routing call requests within a SIP network [9].

3.6.5 SIP Drawbacks

The use of SIP provides a very functional VoIP solution. The network is able to support and provide a wide range of multimedia applications. It has been speculated, the use of text-based messages could add additional message length overhead and processor costs but no study had been done to prove this speculation as of October 2000 [9]. However, network capacity and host capabilities will likely render this argument moot.

Also all reachable routes must be manually provisioned in the proxy and gateway. In a medium to large-scale implementation, the manual provisioning of the same prefix information twice (gateway + proxy) would be costly and possibly prohibitive. Additionally, the proxy has no way to know the dynamic state of the gateway. The lack of dynamic resource information could cause added call blocking to the SIP network.

Another area that could use improvement is using DNS to route SIP requests. The introduction of a dynamic routing protocol responsible for providing the optimum path for session routing would strengthen the performance of the location server and the overall SIP services network. A dynamic protocol, Telephony Routing over IP, is being developed to fill this need and it will be detailed in the next section.

3.7 Telephony Routing over IP (TRIP)

3.7.1 TRIP

As described in the previous section, a SIP network is a service architecture. Its simple text based messaging provides a straightforward communication scheme across an IP network. The scheme is used to support varied multimedia applications including voice over IP services. Also detailed in the previous section was a logical entity referred to as the Location Server (LS). The function of the LS is to provide next-hop routing information for incoming session requests. The LS currently has no way to make routing decisions based on dynamic network resource information. That inability was the reason for development of Telephony Routing over IP (TRIP). TRIP is a routing protocol that runs in conjunction with a SIP/IP network. The task of TRIP is to build a routing table for the proxy it supports. The proxy will utilize that routing table to make session request forwarding decisions.

All TRIP communications are sent across reliable transport (generally TCP). This eliminates the need to implement explicit fragmentation, retransmission, acknowledgment, and sequencing in TRIP. The error notification mechanism used in TRIP assumes that the transport protocol supports a graceful close [12]. TRIP is independent of the underlying VoIP signaling protocol. For example, TRIP can be implemented on a H.323 [13] network as well as a SIP network. H.323 is an ITU standard that provides a foundation for audio, video, and data communications. H.323 defines a unified system for providing multimedia applications. H.323 does not have a RFC. This description will focus on a TRIP implementation over a SIP/IP network.

The physical architecture of a TRIP network is identical to a SIP network. The difference between the two is a TRIP-enabled SIP network includes added clients and applications running on the physical SIP devices/entities. The major entities in a TRIP network are the proxy running a TRIP-enabled location server and the media gateway running a TRIP-lite client. A TRIP-enabled location server is referred to as a TRIP speaker because it messages other entities with TRIP messaging. The location server functionality can be further segregated into a border TRIP speaker and a TRIP speaker

internal to an administrative domain. Each entity of the TRIP architecture will be detailed.

3.7.2 TRIP-lite and the SIP Media Gateway

The TRIP-lite (also called TRIP-GW) client runs on the media gateway. The TRIP-lite client is responsible for advertisement of routes and PSTN prefix destinations reachable through its PSTN trunks. TRIP-lite advertises these routes and prefix destinations to at least one location server. If proxy redundancy is built into the TRIP/SIP network, the TRIP-lite client will advertise the routes and prefix destinations to two or more location servers. Thus, multiple proxy servers would be able to route calls to that single gateway. This eliminates the possibility that a failed proxy server will also remove from service all the gateways it supports. A normal implementation would have each gateway advertise its routes and prefix destinations to a primary location server and a secondary location server. Since TRIP-lite automatically advertises reachable routes to the location server, no manual configuration is required on the proxy. This resolves the proxy manual configuration drawback of a SIP network.

The TRIP-lite client continually updates the location server with dynamic resource information. The types of attributes messaged are destination prefixes, capacity to each prefix destination, dynamic utilization of each trunk group and other statistics usable by the location server to determine the optimum gateway for the next call request. If a specific location server has two gateways, each with a trunk group to one destination prefix, the LS can use the dynamic resource information to load balance across the two gateways. The TRIP-lite dynamic resource messaging resolves the issue of a proxy not having real-time resource knowledge of the SIP network.

Figure 6 shows a TRIP-lite architecture and can be used to illustrate the use of TRIP-lite. Both GW1 and GW3 have routes to the destination prefix 913. The TRIP-lite messaging from each gateway would provide that information along with utilization statistics to the location server. Thus, when a 913 destined call request arrives at the SIP proxy, the LS would be able to route the call to either gateway. And based on the utilization statistics, the LS would be able to choose the GW1 with the lowest 913-trunk utilization. Additionally, if a failure occurs on GW1 and the 913-trunk group goes out of service, the TRIP-lite client would immediately update the location server and all subsequent 913 destined call requests would be routed to GW3 for termination. Once the out of service trunk on GW1 is restored, a TRIP-lite update would be sent and the location server would be able to route to or load balance across both GW1 and GW3.

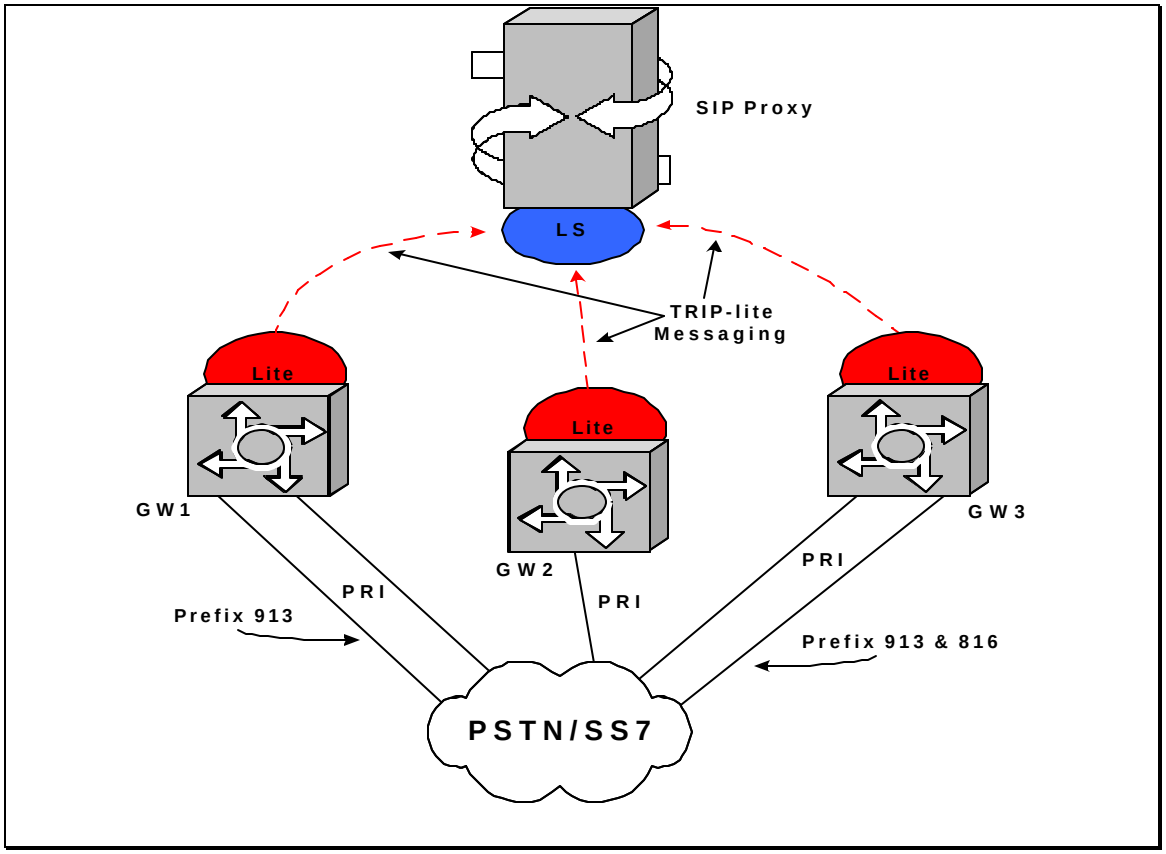

Figure 6: TRIP-lite Messaging to Location Server

3.7.3 TRIP: Interior Administrative Domain Routing (I-TRIP)

As described earlier the functionality provided a location server by TRIP can be divided into two distinct parts. They are TRIP routing within an administrative domain (I-TRIP) and TRIP routing between domains (E-TRIP). A TRIP administrative domain is referred to as an IP Telephony Administrative Domain (ITAD).

The function of I-TRIP is an inter-ITAD gateway location and routing protocol [12]. The primary function of a location server running TRIP, referred to as a TRIP speaker, is to exchange route table information with other location servers. This information includes the reachability of telephony destinations, the routes towards these destinations, and information about gateways towards those telephony destinations residing in the PSTN. The I-TRIP database update messaging is flooded via reliable intra-flooding mechanism modeled after that of the Open Shortest Path First (OSPF). As stated earlier the flooding is made reliable by the transport protocol on which TRIP is supported [12]. Figure 7 shows an I-TRIP architecture showing the flooding process.

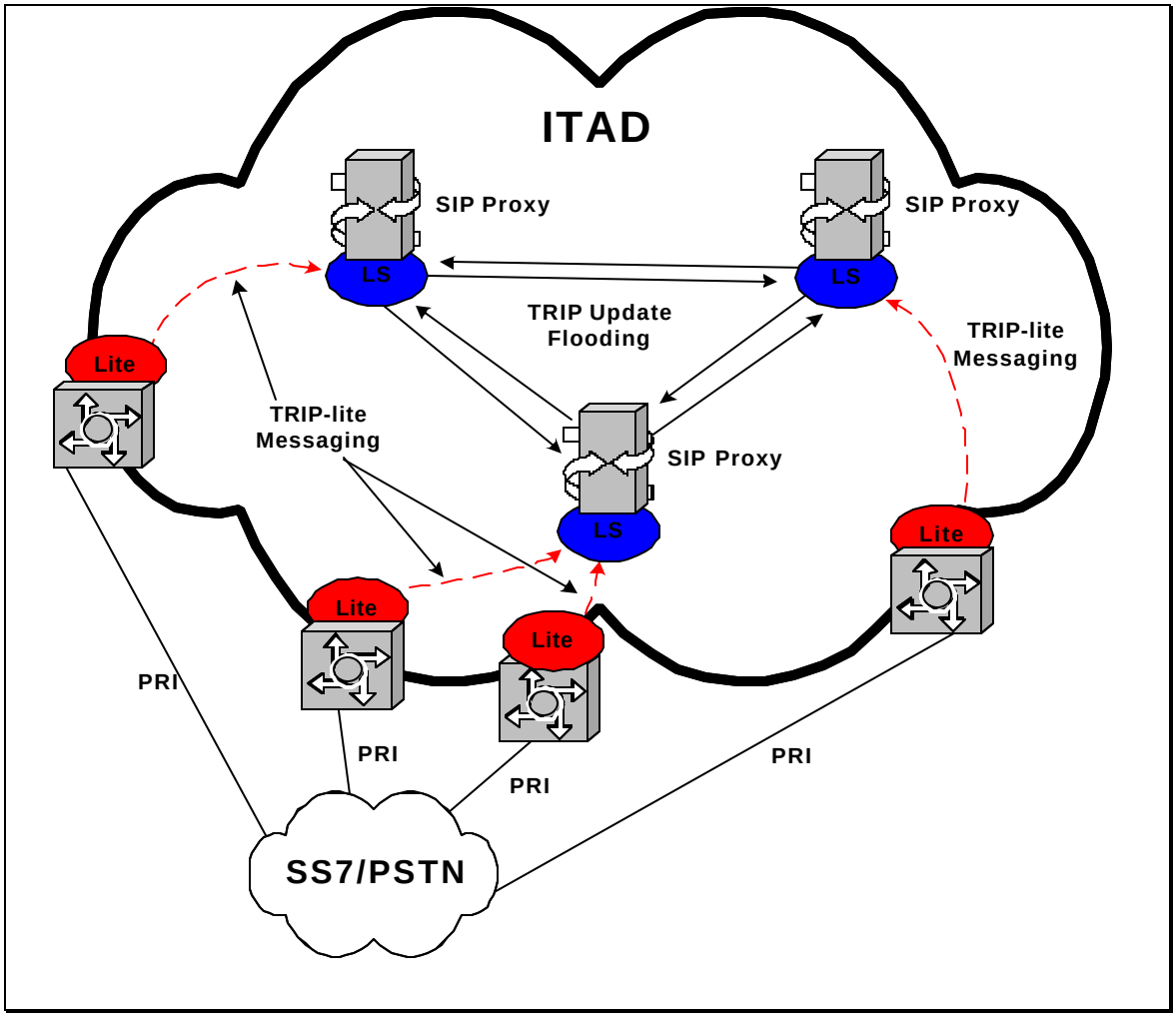

Figure 7: TRIP Routing Updates Inside an Administrative Domain

A peer transport connection is established between two location servers. They exchange messages to open and confirm the connection parameters, and to negotiate the capabilities of each LS as well as the type of information to be advertised over this connection. Keep-alive messages are transmitted throughout the life of the connection.

After initial peer connection setup, the two location servers will exchange their full routing tables. For I-TRIP this includes both internal and external route table information. After the initial table exchange the two peers will only send updates. Theses updates are flooded throughout the ITAD. Once all location servers have received all updates, the internal routing tables (called LOC-TRIB) for all location servers should converge to be identical. This convergence is referred to as synchronization.

When a location server receives an update message, the routes in the update are checked to determine if they are newer than the version already in the database. If newer, the LS will update its route table and then flood that update to all other peers in the same

domain. As stated when all peers in the domain have received that flooded update and made the route table update, the system is synchronized.

TRIP routes are advertised between a pair of location servers in UPDATE messages. The destination addresses and other attributes such as path or egress gateway are included in the message [12].

TRIP allows the SIP network to reroute calls to other proxy servers based on dynamic information. For example, if a gateway is at full utilization of a specific trunk group the TRIP-lite client would message the LS with that dynamic resource information. If that proxy then receives a call request for that destination truck the proxy would know it must send the call request elsewhere for termination. It would then look at its route table and identify a second proxy with a gateway trunk to the specified destination. The primary LS would subsequently reroute the call request to the secondary proxy for termination through its gateway. The next section will discuss E-TRIP, the exterior routing function of TRIP.

3.7.4 TRIP: Exterior Administrative Domain Routing (E-TRIP)

The previous section discussed I-TRIP and its responsibility for distribution of routing information between TRIP speakers in one administrative domain. TRIP also was developed to exchange telephony routing information between administrative domains. This functionality is referred to as E-TRIP.

As discussed earlier, I-TRIP uses reliable flooding to synchronize the routing tables of all TRIP speakers in an ITAD. E-TRIP was developed to function like Border Gateway Protocol Version 4 (BGP-4). TRIP designers actually built the protocol using BGP's inter-domain transport mechanism, BGP's peer communication, BGP's finite state machine, and similar formats and attributes as BGP [12].

E-TRIP peers establish point-to-point links and provide route updates based only on the external routing table (Ext-TRIB). Specific internal routing information is not updated beyond the boundary of the ITAD. Figure 8 shows two ITADs and E-TRIP communication between the two border location servers.

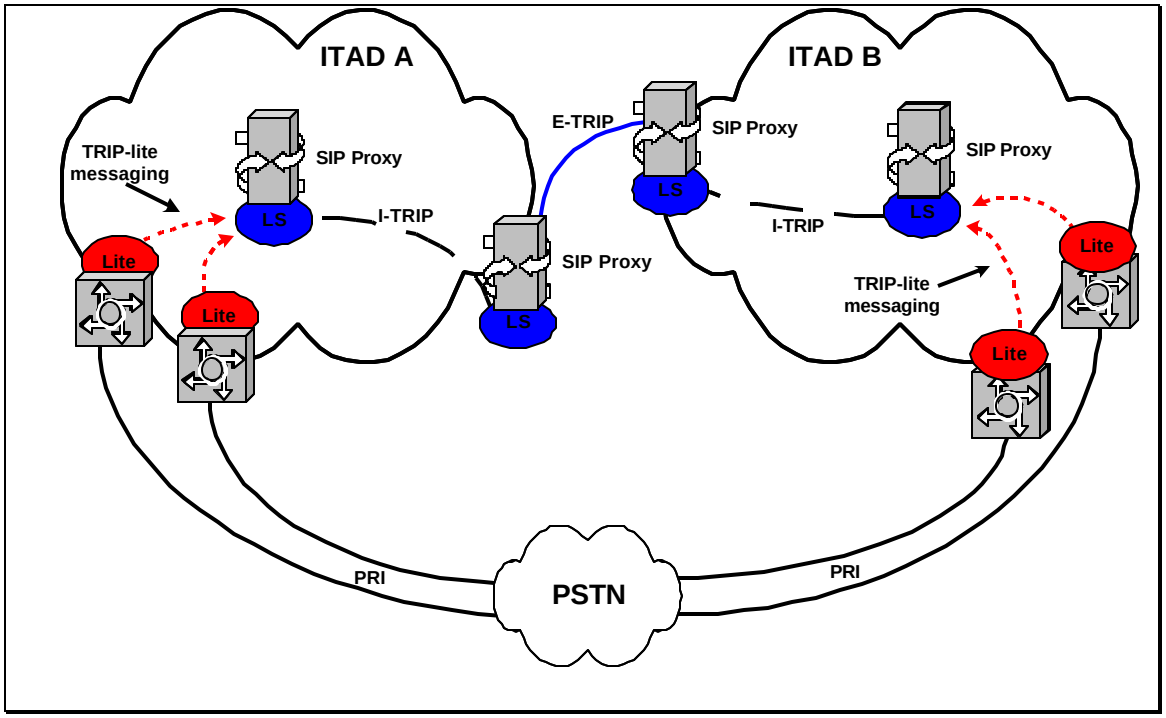

Figure 8: TRIP Routing Updates Between Two Administrative Domains

The remainder of E-TRIP functionality is identical to I-TRIP. This includes routs being transmitted in SIP UPDATE messages and the attributes advertised [12]. I-TRIP and E-TRIP combined provide the resolution for the final issue with a SIP VoIP network. That being the use of DNS lookups to identify next hop information. I-TRIP and E-TRIP provide a dynamic telephony routing protocol. A synchronized TRIP-enabled SIP network should be able to make optimum next hop route decisions.

3.7.5 TRIP Research Issues

The appeal of TRIP is that it immediately solves the three major drawbacks of a SIP network. Those being manual configuration of proxy route tables, lack of dynamic resource knowledge at the gateway, and use of DNS for identification of the next hop. However, there are open issues about a TRIP-enabled SIP network. The questions revolve around the impact of varying physical parameters associated with a TRIP-enabled SIP network. The questions are:

- ?? What is the impact of location server-to-gateway propagation delay and location server to location server delay on blocking probability in a TRIP environment?
  - o The importance of this issue to a carrier concerns deployment of location servers to support the network. Determining propagation delay impact on call blocking will allow designers to place the fewest number of location servers to support demand in a given geographic area.
- ?? What is the impact of location server-to-gateway delay and location server to location server delay on call request delivery in a TRIP environment?

- The importance of this issue to a carrier concerns quality of service provided to customers. Determining where delay will be incurred will allow designers to meet delay budget.

?? What is the impact of location server-to-gateway delay and location server to location server delay on location server call request rerouting in a TRIP environment?

- The importance of this issue to a carrier concerns call setup. Determining the impact of call reroutes on call setup delay impacts the delay budget.

?? What is the impact of traffic intensity along with location server-to-gateway delay and location server to location server delay variation in a TRIP environment?

- The importance of this issue to a carrier is to understand how the network will react under varying load conditions. A network designer would be able to design the network to a specific load.

?? What is the impact of trunk failure along with location server-to-gateway delay and location server to location server delay variation on blocking probability in a TRIP environment?

- The importance to a carrier is to understand how a failure will impact customers.

?? How does the system blocking performance of a TRIP network differ from the blocking performance of a SIP network?

- The importance to a carrier is to understand how the addition of TRIP will affect a SIP network. The addition of TRIP will add complexity and cost over a SIP network. This experiment will show the benefits.

These questions are the basis for investigation performed using a simulation model of a TRIP-enabled SIP network. The following two sections provide the test plan executed and a description of the model employed.

4.0 TRIP Evaluation Test Plan

4.1 Test Plan Objective

The purpose of this test plan is to evaluate the performance of voice over IP (VoIP) network utilizing TRIP and TRIP-lite. The plan calls for results based on call blocking probabilities, call request delivery time and percentage of call request reroutes between location servers. The plan is aimed at evaluating the impact of varying network topology via location server-to-gateway propagation delay, location server to location server propagation delay and interarrival rate of call requests in a TRIP-enabled network. Additionally, system impacts during gateway trunk failure and recovery will be studied. Finally, the model will provide results based on a SIP network without TRIP and TRIP-lite. The SIP results will be compared to the TRIP results.

The goal will be to obtain data that can be used to draw conclusions that answer the questions posed earlier in the TRIP background section. Again, the questions are:

- What is the impact of location server-to-gateway delay and location server to location server delay on blocking probability in a TRIP environment?
- What is the impact of location server-to-gateway delay and location server to location server delay on call request delivery in a TRIP environment?
- What is the impact of location server-to-gateway delay and location server to location server delay on location server call request rerouting in a TRIP environment?
- What is the impact of traffic intensity along with location server-to-gateway delay and location server to location server delay variation in a TRIP environment?
- What is the impact of trunk failure along with location server-to-gateway delay and location server to location server delay variation on blocking probability in a TRIP environment?
- How does the system blocking performance of a TRIP network differ from the blocking performance of a SIP network?

Below are the system variables to be varied and the system performance metrics that will be recorded to support conclusions.

Varied System Parameters	**System Performance Metrics**
- LS-to-GW Propagation Delay	- Overall System Call Blocking
- LS-to-LS Propagation Delay	- LS Call Blocking
- Traffic Load	- GW Call Blocking
- GW Trunk Failure	- Call Request Delivery Delay
	- % LS Call Request Rerouting
	- Cumulative Blocked Calls during Gateway Trunk Failure

4.2 Model Evaluation

The TRIP simulation model is based on a specific network configuration as well as several assumptions. They are listed below.

1. Two Locations Servers running TRIP.
2. Each LS has a single gateway each with trunks to the destination prefix.
3. Each gateway is running the TRIP-lite client.
4. A carrier wants blocking to occur at the LS. This will allow the primary LS to reroute the call to the secondary LS and allow the secondary LS to complete the call through its gateway.
5. The rerouting of call requests between location servers is controlled by SIP. The SIP RFC dictates a call request not be routed to a single location server more than once. Thus, a single location server will encounter any individual call request once.
6. The rerouting of call requests is only handled between location servers. When a reroute is required, the initial location server will route the call request to the secondary location server not directly to the secondary gateway.
7. TRIP-lite messaging delay between the gateway and LS causes the LS to become unsynchronized with the dynamic resources on the gateway. This causes blocking at the gateway. As the delay increases, the gateway blocking increases.
8. A carrier wants the lowest possible call request delivery time.
9. Call request rerouting between location servers cause increased call request delivery time.
10. At most two location servers, a primary and secondary, will control gateways with the same destination prefix. This will allow for one call request reroute.
11. A call request will be blocked when both primary and secondary location servers block the call.
12. Design of the IP/SIP network forces packet loss due to TCP/IP and network congestion to be insignificant.
13. Throughout all simulations average call holding time is three (3) minutes.
14. Throughout all simulations control message length is 4096 bits.
15. Throughout all simulations link capacity is 100Mb/second.

4.3 Configuration Evaluation

A) The motivation for variation of LS-to-gateway propagation delay and variation of traffic intensity is to provide understanding of how physical separation of LS and GW impact system performance.
    1) Vary the LS-to-gateway propagation delay.
        i. The lower limit on the sensitivity analysis will be delay set to near zero (0.01ms), LS and gateway co-located in same central office.
        ii. The upper limit will be the propagation delay incurred traversing a fiber run across the continental United States. The distance between

New York and San Diego is 3000 miles. This is equivalent to a propagation delay of 24msec.

iii. Additionally, two runs will be preformed with a delay of 125ms and 250ms. The 250ms run will simulate the use of a satellite link.

iv. LS-to-LS delay will be set to 4msec (500 miles) for this analysis.

2) Plot blocking probability versus time with each delay curve plotted. The resulting plot will show the effect of increasing the LS-to-gateway delay.
   i. The blocking probability versus time analysis will vary the propagation delay. The three delay values will be the following:
      1. 0.01msec: co-located
      2. 24msec: cross country fiber connection
      3. 250msec: satellite link.
   ii. The dynamics of system call blocking, LS call blocking, GW call blocking, call request delivery time and call request reroute percentage will be presented.
3) Steady state values for system blocking, LS blocking, GW blocking, call request delivery time and call request reroute percentage will be estimates as function of propagation delay. Plot the steady state values versus LS-to-gateway propagation delay.
   i. Plot system blocking, LS blocking, GW blocking, and predicted Erlang B.
4) Plot call request delivery time versus time for each LS-to-gateway propagation delay.
   i. Use results to plot steady state call request delivery time versus LS-to-gateway propagation delay.
5) Plot call request reroute percentage versus time for each LS-to-gateway propagation delay.
   i. Use results to plot steady state call request reroute versus LS-to-gateway propagation delay.
6) Plot the percentage of GW call blocking to system blocking versus LS-to-gateway propagation delay. This will graphically depict the relative percentage of call blocking at the GW to overall system blocking.
   i. Use results to plot steady state call request reroute versus LS-to-gateway propagation delay.
7) Repeat analysis ten (10) times, record all results and average. All plots will be generated from the average steady state values. Repeating the analysis will provide the basis for confidence bounds on this experiment.
8) Repeat steps 1-6 and vary interarrival time to vary traffic intensity. The analysis will vary interarrival time to generate 1%, 5%, 15%, 35%, 65%, and 85% system blocking.
   i. Plot system blocking, LS blocking, GW blocking versus interarrival time.
   ii. Plot call request delivery time versus interarrival time.
   iii. Plot call request reroute versus interarrival time.

B) The motivation for variation of LS-to-LS propagation delay and variation of traffic intensity is to provide understanding of how physical separation of LS and LS impact system performance.
    1) Vary the LS-to-LS propagation delay.
        i. The lower limit on the sensitivity analysis will be delay set to near zero (0.01ms), both location servers co-located in same central office.
        ii. The upper limit will be the propagation delay incurred traversing a fiber run across the continental United States. The distance between New York and San Diego is 3000 miles. This is equivalent to a propagation delay of 24msec.
        iii. Additionally, two runs will be preformed with a delay of 125ms and 250ms. The 250ms run will simulate the use of a satellite link.
        iv. LS-to-gateway delay will be set to 4msec (500 miles) for this analysis.
    2) Plot blocking probability versus time with each delay curve plotted. The resulting plot will show the effect of increasing the LS-to-LS delay.
        i. The blocking probability versus time analysis will vary the propagation delay. The three delay values will be the following:
            1. 0.01msec: co-located
            2. 24msec: cross country fiber connection
            3. 250msec: satellite link
        ii. The dynamics of system call blocking, LS call blocking, GW call blocking, call request delivery time and call request reroute percentage will be presented.
    3) Steady state values for system blocking, LS blocking, GW blocking, call request delivery time and call request reroute percentage will be estimates as function of propagation delay. Plot the steady state values versus LS-to-LS propagation delay.
        i. Plot system blocking, LS blocking, GW blocking, and predicted Erlang B.
    4) Plot call request delivery time versus time for each LS-to-LS propagation delay.
        i. Use results to plot steady state call request delivery time versus LS-to-LS propagation delay.
    5) Plot call request reroute percentage versus time for each LS-to-LS propagation delay.
        i. Use results to plot steady state call request reroute versus LS-to-LS propagation delay.
    6) Plot the percentage of GW call blocking to system blocking versus LS-to-LS propagation delay. This will graphically depict the relative percentage of call blocking at the GW to overall system blocking.
        i. Use results to plot steady state call request reroute versus LS-to-LS propagation delay.
    7) Repeat analysis ten (10) times and record all results. Repeating the analysis will provide the basis for confidence bounds on this experiment.

8) Repeat steps 1-6 and vary interarrival time to vary traffic intensity. The analysis will vary interarrival time to generate 1%, 5%, 15%, 35%, 65%, and 85% system blocking.
    i. Plot system blocking, LS blocking, GW blocking versus interarrival time.
    ii. Plot call request delivery time versus interarrival time.
    iii. Plot call request reroute versus interarrival time.

C) The motivation for variation of LS-to-gateway along with introduction of trunk failure is to provide understanding of how physical separation of LS and GW teamed with a gateway trunk failure impact system performance.
    1) Allow system to achieve steady state. This depends on the interarrival rate. The greater the interarrival rate, the quicker the system will achieve steady state.
    2) Once the system arrives at steady state, simulate loss of 24 of 48 trunks from gateway 1.
    3) Once the system again arrives at steady state, simulate correction of trunk problem and return to 48 trunks on gateway 1.
    4) Plot number of blocked calls versus time. The sensitivity analysis will overlay each delay curve as the LS-to-gateway propagation delay is increased. The resulting plot will show the effect of increasing the LS-to-GW delay.
        i. The number of blocked calls versus time analysis will vary the propagation delay. The three delay values will be the following:
            1. 0.01msec: co-located
            2. 24msec: cross country fiber connection
            3. 250msec: satellite link
        ii. Number of blocked calls will be plotted versus time. The plots will be broken into three separate intervals: before failure, after failure and before restoral, and after restoral. The reason for the split is to look at each interval individually and understand the system reaction due to the change in system state.

D) The motivation for variation of LS-to-LS along with introduction of trunk failure is to provide understanding of how physical separation of LS and LS teamed with a gateway trunk failure impact system performance.
    1) Allow system to achieve steady state. This depends on the interarrival rate. The greater the interarrival rate, the quicker the system will achieve steady state.
    2) Once the system arrives at steady state, simulate loss of 24 of 48 trunks from gateway 1.
    3) Once the system again arrives at steady state, simulate correction of trunk problem and return to 48 trunks on gateway 1.
    4) Plot number of blocked calls versus time. The sensitivity analysis will overlay each delay curve as the LS-to-LS propagation delay is increased. The resulting plot will show the effect of increasing the LS-to-LS delay.

i. The number of blocked calls versus time analysis will vary the propagation delay. The three delay values will be the following:
    1. 0.01msec: co-located
    2. 24msec: cross country fiber connection
    3. 250msec: satellite link
ii. Number of blocked calls will be plotted versus time. The plots will be broken into three separate intervals: before failure, after failure and before restoral, and after restoral. The reason for the split is to look at each interval individually and understand the system reaction due to the change in system state.

E) The motivation for comparing a TRIP network to a SIP network is to understand the benefits provided by the implementation of TRIP. Also, to understand the performance of a SIP network with no TRIP-lite dynamic resource messaging or LS-to-LS rerouting.
    1) Disable TRIP-lite messaging from both GW1 and GW2.
    2) Apply LS-to-GW delay of 4ms.
    3) The analysis will vary interarrival time to generate 1%, 5%, 15%, 35%, 65%, and 85% system blocking.
        i. Plot system blocking versus interarrival time.
    4) Plot SIP simulation results on applicable graphs to compare to TRIP simulation results.
F) Use analysis in [15] to locate a confidence interval for 1% blocking. The 1% blocking case will have the fewest events (calls) and so it will have the least stringent confidence bounds.

4.4 Expected Trends:

There are several expected results associated with the test plan. They are listed below.

1. When all model delays are set to near zero the model blocking probability will approximate Erlang B given specified holding time and interarrival rate.
2. Given the number of trunks remains constant, as interarrival time increases blocking probability will increase.
3. As interarrival time increases, call request delivery time will increase.
4. As interarrival time increases, call request reroutes will increase.
5. As location server-to-gateway delay increases, location server blocking will decrease and gateway blocking will increase.
6. As location server to location server delay increases, call request delivery time will increase.
7. Trunk failure in one gateway will drive increased blocking in both gateways.
8. The TRIP-enabled simulations will provide lower system blocking at each interarrival rate than those provided by the SIP simulations.

The following section will describe in detail the performance model used to execute this test plan.

5.0 TRIP Simulation Model Description

5.1 Model Purpose:

The purpose of this model is to simulate the signaling and blocking performance of a voice over Internet Protocol (VoIP) network utilizing TRIP and TRIP-lite. The model will provide results based on call blocking probabilities, call request delivery and percentage of call request reroutes between location servers. The model will be used to evaluate the impact of varying location server-to-gateway propagation delay, location server to location server propagation delay and interarrival rate of call requests. Additionally, the model will be utilized to evaluate system impacts during gateway trunk failure and recovery.

The model will provide call blocking results for the overall system and the individual call blocking at location servers and at gateways. The results of location server and gateway blocking probability will provide an understanding of where call blocking is occurring as the system parameters are varied. From a carrier point of view, gateway call blocking is unwanted. A carrier wants blocking to occur at the location server. This will allow the primary location server to reroute the call to the secondary location server and allow the secondary location server to complete the call through its gateway. The model will also provide call request delivery time results. The call request delivery time will be calculated as the difference between call request origination and delivery to a gateway for call termination.

The final result evaluated is call request rerouting between the two location servers. The inclusion of TRIP in this model provides each location server an alternate route to the destination prefix. When the primary location server is alerted that its gateway is at full trunk capacity, the primary location server will route the call request to the secondary location server. The percentage of call request reroutes will be evaluated as the system parameters are varied.

5.2 Model Description:

The TRIP model was built using Extend Version 5 [14]. The model is logically built with two location server/gateway pairs. Each pair functions independently of the other and each is fed call requests by a dedicated call request generator. Location server 1 (LS1) is able to send calls to only Gateway 1 (GW1). Accordingly, LS2 is only able to send calls to GW2. When a call is sent from LS to its GW, it passes through a delay block that represents the propagation delay incurred due to their physical separation. This propagation delay is a physical characteristic of the model and variation of its size impacts the performance of the model. Figure 9 is a high level illustration of the model architecture.

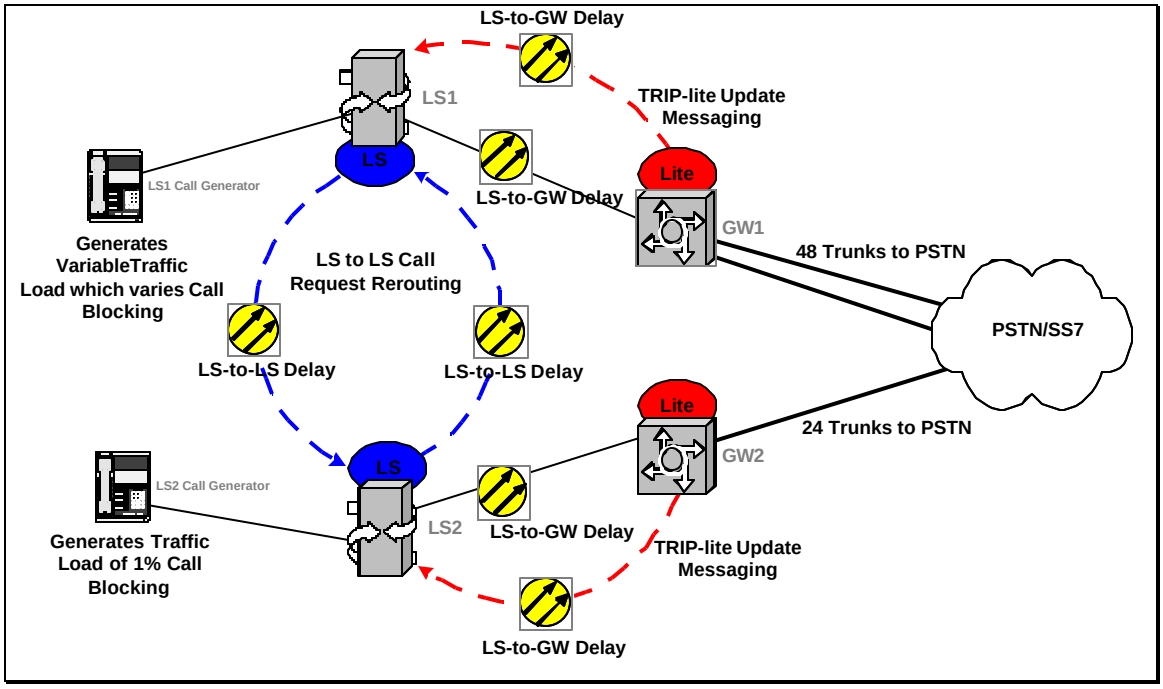

Figure 9: High Level Model Architecture

Each GW is built to simulate its running a TRIP-lite client. This TRIP-lite client provides a feedback loop from the GW to a decision block located in front of the corresponding LS. The TRIP-lite feedback loop is designed to allow the GW to notify the LS when all its trunks are busy. The TRIP-lite messaging (feedback loop) traverses the same physical separation as the initial call so the messaging also includes a delay block to represent propagation delay. Again, this propagation delay impacts the performance of the model. Vendor implementations of TRIP-lite would provide additional messaging which would allow the location server to have greater knowledge of gateway resources. This model simulates the worst-case scenario where the location server is only updated when the gateway is at full capacity.

After the LS decision block is notified its GW is at full trunk utilization, call requests will be rerouted to the secondary LS. When the GW has open capacity call requests will be sent to the GW. The rerouting between location servers simulates the inclusion of TRIP in the model. As described earlier in this document, TRIP is used to build LS routing tables. In this model it is assumed that TRIP communication would have built each LS routing table such that each LS would have knowledge of another gateway with trunks to the required destination and the gateway was reachable through the secondary LS.

When a call request is rerouted from one LS to the secondary LS, it passes through a delay block that represents the propagation delay incurred due to physical separation of the two location servers. The call request will then enter a decision block for the secondary LS. If the secondary GW has open capacity the call will be sent to the GW. If the secondary GW is at full capacity, the call will be blocked. The following section will describe in detail each major block of the TRIP model.

5.3 Description of Model elements in Simulation Model

LS1 Call Generation

The LS1 Call Generation block is responsible for originating call requests to load the LS1/GW1 pair. This block is where the model traffic intensity is varied. This is accomplished by varying the interarrival rate. Additionally, each call originated is given a value of 1, which is used in the LS-to-LS Delay and Blocking Decision section. Finally, each call request is stamped with the time it was originated. This time stamp will be used to evaluate call setup time. Figure 10 shows the LS1 call generator hierarchical block view as well as the interior of the hierarchical block.

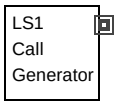

Figure 10a: Interior of Call Generation Hierarchical Block

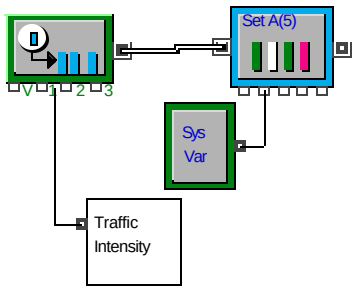

Figure 10b: Call Generator

LS2 Call Generation

The LS2 Call Generation block is responsible for originating call requests to load the LS2/GW2 pair. This block is configured to provide LS2/GW2 with a nominal constant traffic intensity of 1% or fifteen (15) Erlang. As in the LS1 call generation block, each LS2 call request originated is given a value of 1 and stamped with the time it was originated. Figure 10 shows the LS2 call generator hierarchical block view as well as the interior of the hierarchical block.

LS1 TRIP-lite Decision

The LS1 TRIP-lite decision block is responsible for determining if a call should be routed to the gateway for call termination or if the call request should be rerouted to the secondary LS. This decision is made utilizing the feedback generated by the simulated TRIP-lite client running on GW1. The GW1 TRIP-lite client will message LS1 when all trunks are busy. When the decision block receives this message it reroutes the call request through its b connector to LS-to-LS Reroute delay section. Figure 11 shows the LS1 TRIP-lite decision block.

This block represents the use of TRIP-lite dynamic resource messaging at the LS to decide if an alternate path should be used.

To simulate a SIP network, the select will be supplied with a constant value of zero (0). This disables the TRIP-lite messaging from the GW1 to LS1 Update section.

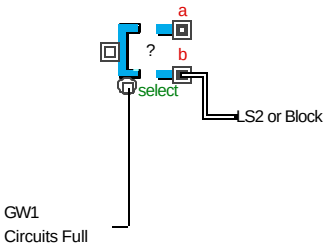

Figure 11: TRIP-lite Decision Block

LS2 TRIP-lite Decision

The LS2 TRIP-lite decision block is identical to LS1 decision block. The only difference is that it receives dynamic resource messaging from GW2. Figure 11 shows the LS2 TRIP-lite decision block. As with the LS1 TRIP-lite decision block, this block represents the use of TRIP-lite dynamic resource messaging at the LS to decide if an alternate path should be used.

LS1 & LS2 SIP Proxy

The SIP proxy uses the link capacity and packet length to determine service time. In this simulation link capacity and packet length are not varied. Thus, the service time of the SIP proxy remains constant. The secondary responsibility of the proxy is to count the number of calls routed to each gateway. The number of calls sent to each gateway is used to calculate the associated gateway blocking probability. Figure 12 shows the LS1 and LS2 SIP proxy hierarchical block view as well as the interior of the hierarchical block.

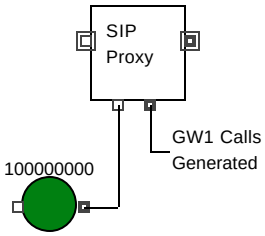

Figure 12a: Interior of LS1 and LS2 SIP Proxy Hierarchical Block

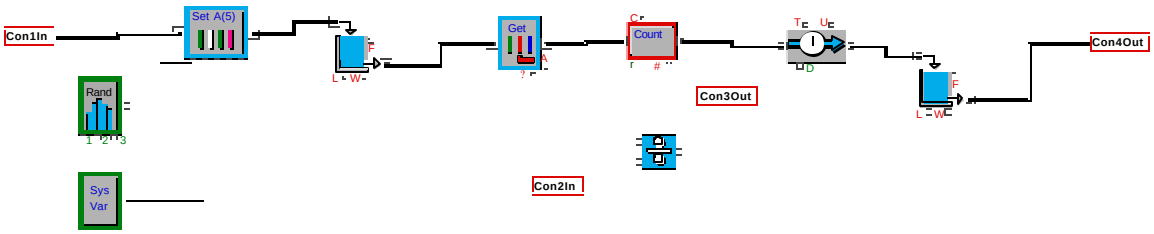

Figure 12b: LS1 and LS2 SIP Proxy

LS-to-GW Delay

The LS-to-GW delay block represents the propagation delay incurred due to the physical separation of the location server and gateway. This delay impacts the performance of the system. As the delay increases the gateway and LS will become increasingly unsynchronized. This lack of synchronization causes calls to be misrouted. Misrouting of calls causes LS-to-LS rerouting that may not have been required and increased call blocking at the gateway. Figure 13 shows the LS-to-GW delay block.

This delay is varied using sensitivity analysis to evaluate the system impact of propagation delay caused by physical separation of the location server and gateway.

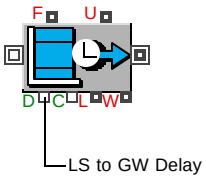

Figure 13: LS-to-GW Delay Block

Call Request Delivery Calculation

The call Request Delivery calculation block is responsible for determining the time delay incurred between call request generation and delivery to a gateway for call termination. The call generation blocks stamp each call request with its originating time. The call request delivery block subtracts the originating time from the current time as a call request passes though and the subsequent value is call request delivery time for that specific call. Additionally, the call request delivery block counts the number of rerouted calls. This value is utilized to determine the percentage of calls rerouted from the primary LS to the secondary LS. Figure 14 shows the call request delivery calculation hierarchical block view and as well as the interior of the hierarchical block.

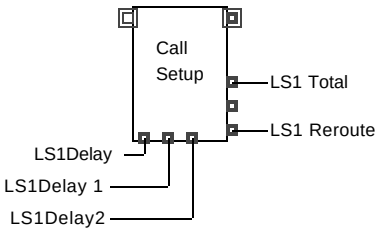

Figure 14a: Interior of Call Request Delivery Hierarchical Block

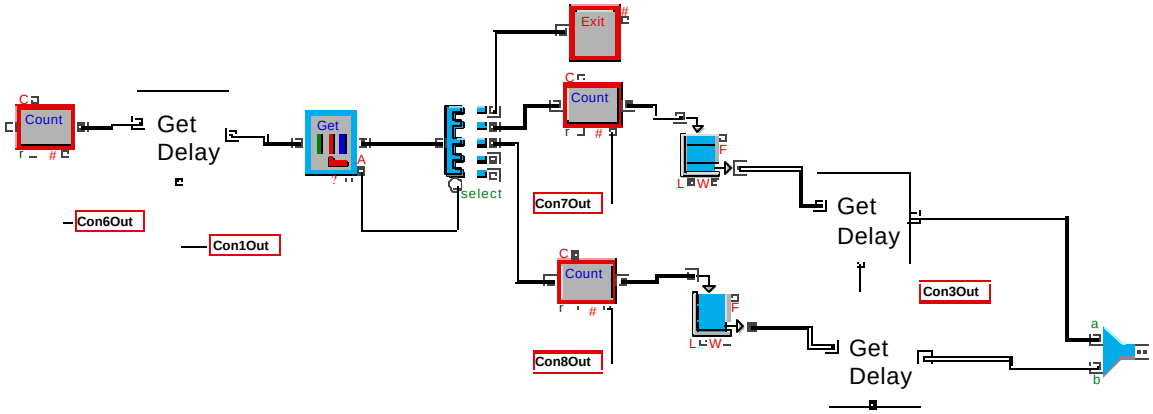

Figure 14b: Call Request Delivery Calculation

GW1: 48 trunks with Failure

The GW1 section is built to simulate a gateway with forty-eight (48) DS-0s. The average holding time for each call is three (3) minutes. This average value is not varied. GW1 also includes logic to cause failure of twenty-four (24) DS-0s. This is accomplished by using an equation and a decision block to circumvent the second set of 24 DS-0s. The GW to LS Update Delay section has a call generator block that changes the trunk attribute at a specified time in the simulation. A failure is triggered when the equation block is notified the GW to LS Update Delay call generator has altered the number of trunks. Restoral of the failed DS-0s is simulated in the same fashion. Figure 15 shows GW1.

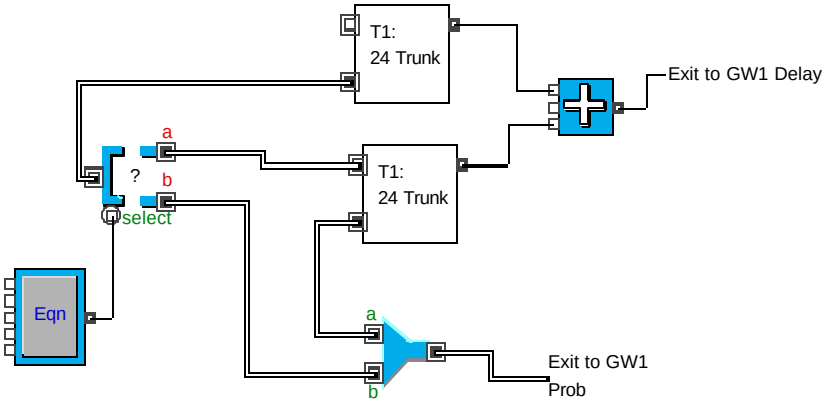

Figure 15: GW1 with Forty-Eight Trunks and Failure

GW2: 24 trunks

The GW2 section is built to simulate a gateway with twenty-four (24) DS-0s. Unlike GW1, GW2 cannot simulate DS-0 failure. Figure 16 shows GW2.

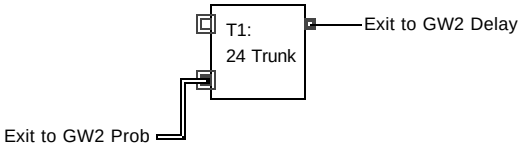

Figure 16: GW2 with Twenty-Four Trunks

GW1 to LS1 Update and Delay

The GW1 to LS1 update and delay section simulates the operation of the TRIP-lite client running on GW1. This block is responsible for dynamic evaluation of trunk utilization on GW1. GW1 is designed to output the number of busy trunks. The GW1 to LS1 update section uses the GW1 utilization information to determine if a message must be sent to the LS1 TRIP-lite decision block. A message will only be sent when GW1 is at full trunk utilization. Figure 17 shows the block components GW1 to LS1 update delay section.

Also, this section includes a delay block that simulates the propagation delay incurred traversing the physical separation between the gateway and location server. As with the LS-to-GW delay, the GW to LS delay will be varied using sensitivity analysis to evaluate the system impact of propagation delay caused by physical separation of the location server and gateway.

Additionally, this section includes a call generator that acts as the catalyst for trunk failure in GW1. At a specified time in the simulation, the call generator changes the number of trunks and this causes the failure or restoral of a set of trunks.

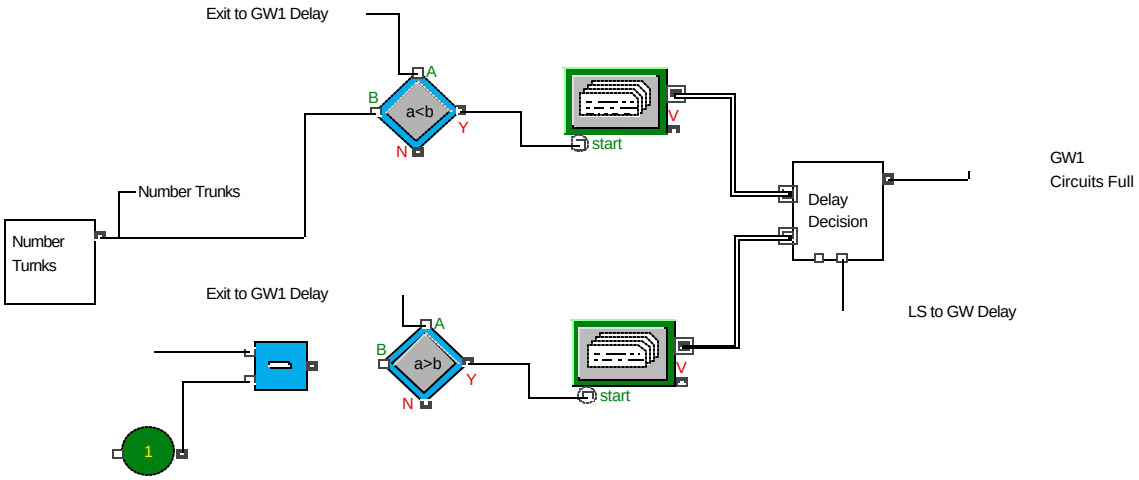

Figure 17: GW to LS Update and Delay

GW2 to LS2 Update and Delay

The GW2 to LS2 update and delay section is identical to the GW1 to LS1 Update and Delay section except that it does not have logic designed to simulate trunk failure. Figure 17 shows the block components.

LS-to-LS Delay and Blocking Decision

The LS-to-LS delay and blocking section is responsible for call request rerouting and delay. The TRIP-lite decision blocks send calls to this section based on TRIP-lite resource updates from the gateways. If the gateway is at full utilization, the call request is rerouted through this section to the secondary LS. Figure 18 shows the block components LS-to-LS delay and blocking decision section.

A simulation assumption is a call request may only be rerouted once. This is based on the premise that a SIP call request be routed to a single location server once. This section will determine if the call request has already been rerouted. This is determined by the value of the call request. At generation, each call request is given a value of one (1). If sent to this section, the value is immediately incremented by one. Thus, after the initial reroute of a call request, its value will be 1 + 1 or 2. If the value is greater than two the call request has been rerouted back to its originating LS and as explained, this is not allowed. Any call request with a value greater than two will be blocked. A blocked call will then be forwarded to the LS blocking calculation section.

This section also is responsible for adding LS-to-LS propagation delay. This propagation delay is incurred due to the physical separation of the primary and secondary location servers. This delay value will also be evaluated using sensitivity analysis. If a call request successfully traverses this section, it is sent to the secondary location sever.

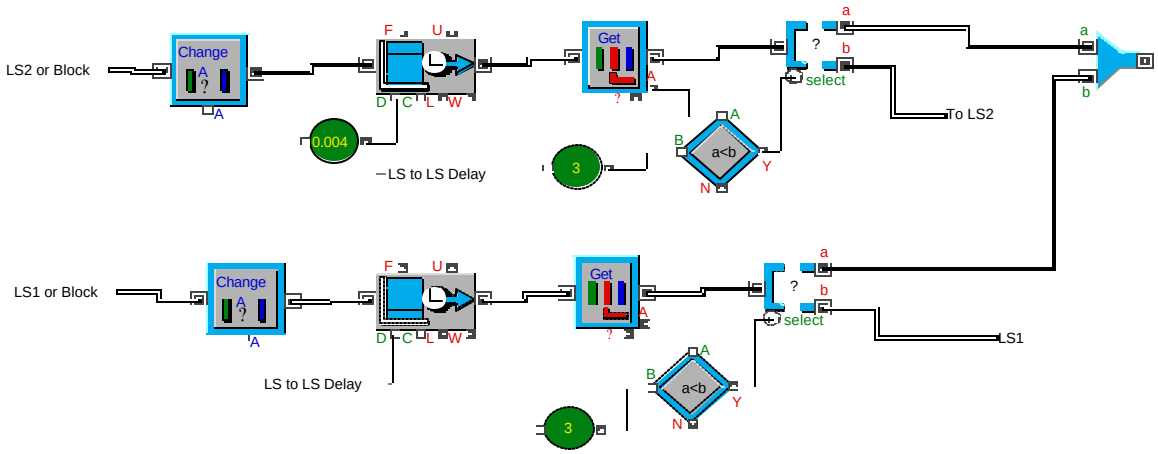

Figure 18: LS-to-LS Delay and Blocking Decision

System Blocking Calculation

The system blocking calculation block is responsible for calculation of overall system blocking probability. It calculates call blocking by summing the number of blocked calls for GW1 + GW2 + total LS calls and dividing by the total number of calls. The subsequent value is multiplied by 100 to provide a percentage. A plotter is used to generate a graph of the blocking probability versus simulation time. Figure 19 shows the block components of the system call blocking calculation.

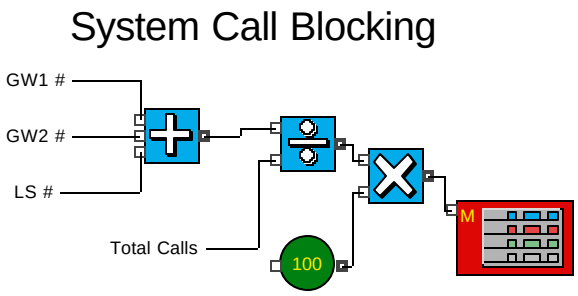

Figure 19: Call Blocking Calculation

LS Blocking Calculation

The LS blocking calculation block is responsible for calculation of location server blocking probability. It calculates call blocking by summing the number of blocked calls for LS1 + LS2 and dividing by the total number of calls. The subsequent value is multiplied by 100 to provide a percentage. A plotter is used to generate a graph of the blocking probability versus simulation time. Figure 19 shows the block components of the LS call blocking calculation.

GW Blocking Calculation

The GW blocking calculation block is responsible for calculation of gateway blocking probability. It calculates call blocking by summing the number of blocked calls for GW1 + GW2 and dividing by the total number of calls. The subsequent value is multiplied by 100 to provide a percentage. A plotter is used to generate a graph of the blocking probability versus simulation time. Figure 19 shows the block components of the GW call blocking calculation.

Overall Call Request Delivery Calculation

The call request delivery time calculation section is responsible for locating the time delay incurred traversing the system from call request origination to call termination at a gateway. The specific call request delivery time of each individual call is determined in the call setup calculation section. Each call delay value is then sent to this section where it is added with call statistics from the other LS/GW pair. The combined value is the overall call request delivery time of the system. A plotter is used to generate a graph of the call request delivery time versus simulation time. Figure 20 shows the block components of the call request delivery time calculation.

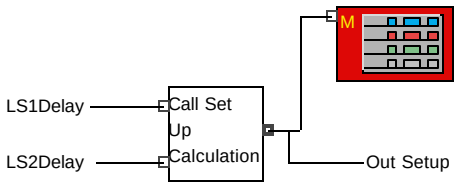

Figure 20: Overall Call Request Delivery Calculation

Call Request Reroute Percentage Calculation

The call request reroute section is responsible for calculating the percentage of call requests rerouted from the primary location server to the secondary location server. This is accomplished by taking the number of reroutes and dividing by the total number of gateway calls. The call request reroute block counts the number of rerouted calls and total number of calls. Both values are sent to this section and the percentage is located. A plotter is used to generate a graph of the call reroute percentage versus simulation time. Figure 21 shows the block components of the call request reroute calculation.

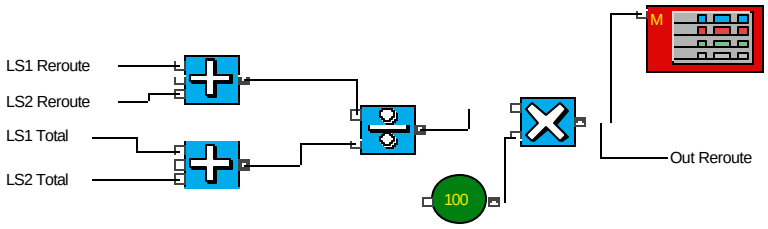

Figure 21: Call Request Reroute Percentage Calculation

Steady State Values Output to Text File

The final values of overall system blocking probability, location server blocking probability, gateway blocking probability, call request delivery time, and call request reroute percentage are all output to a text file. The final values are assumed to be the steady state value. These values are used to plot steady state values for each versus delay and versus interarrival rate. Figure 22 shows the block where steady state values are output to a file.

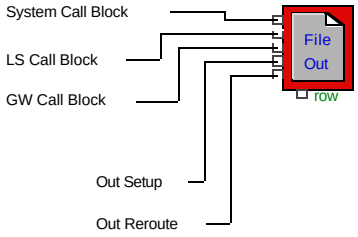

Figure 22: Steady State Values to Output File

The following section will provide the results from the test plan in Section 4. Additionally, conclusions drawn from the results will be provided.

6.0 TRIP Simulation Results and Conclusions

This section will provide the results of the TRIP simulation test plan presented in Section 4.0. The TRIP questions related in Section 4.1 will be addressed and answered. Also, conclusions based on the results will be drawn.

6.1 Impact of Propagation Delay and Interarrival Rate on Blocking Probability

Figure 23 shows the system blocking versus time for an interarrival rate of 57.95 Erlang (1% call blocking). Variable propagation delay is introduced between LS1 and LS2. Figure 24 shows the system blocking versus time for an interarrival rate of and 66.65 Erlang (5% call blocking). Variable propagation delay is introduced between LS and GW. Both Figure 23 and 24 show three propagation delay curves, (0ms, 24ms, and 250ms). The graphs show that delay does not impact blocking as all three delays provide similar curves no matter what delay is introduced or where (e.g., LS-to-LS or LS-to-GW). Additional results were generated for 5%, 35%, 65%, and 85% with LS-to-GW and LS-to-LS delay and all support the conclusion that propagation delay does not impact system blocking.

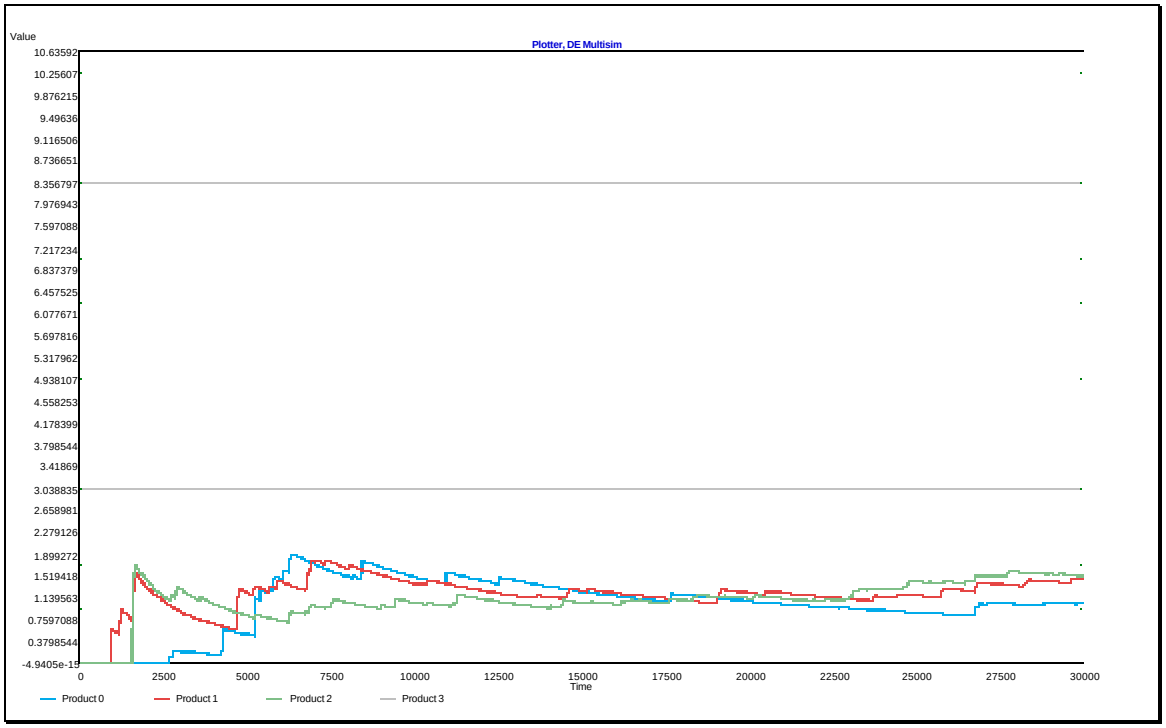

Figure 23: System Call Blocking Value vs. Time, 1% call blocking, LS-to-LS Delay Variation, Blue: 0ms; Red: 24ms; Green: 250ms

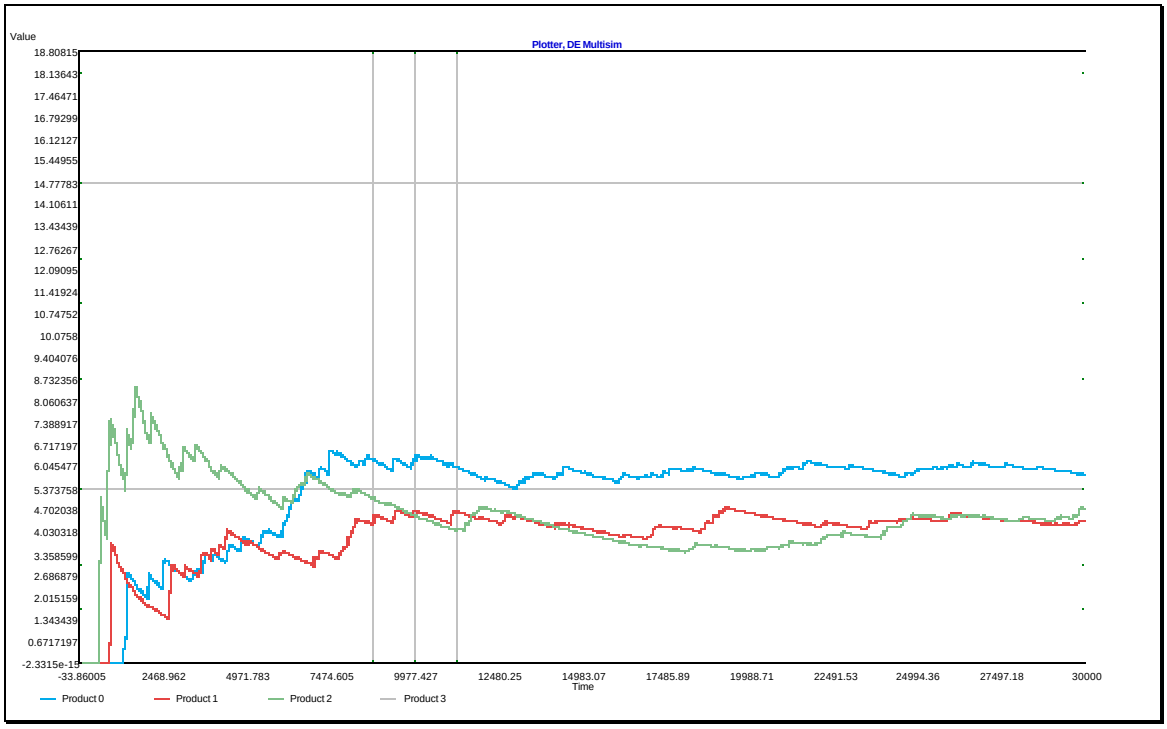

Figure 24: System Call Blocking Value vs. Time, 5% call blocking, LS-to-GW Delay Variation, Blue: 0ms; Red: 24ms; Green: 250ms

Figure 25 shows system blocking delay versus interarrival rate as the LS-to-GW propagation delay is varied. The figure clearly shows system blocking is dependent upon the traffic intensity, as the load is increased the system blocking increases. The figure also includes the predicted Erlang B curve for each interarrival rate. It shows that for every delay introduced (0ms through 250ms), system blocking follows Erlang B. An identical result was generated for LS-to-LS delay variation and it also supports the conclusion that a in TRIP-enabled network system blocking is mainly driven by traffic load.

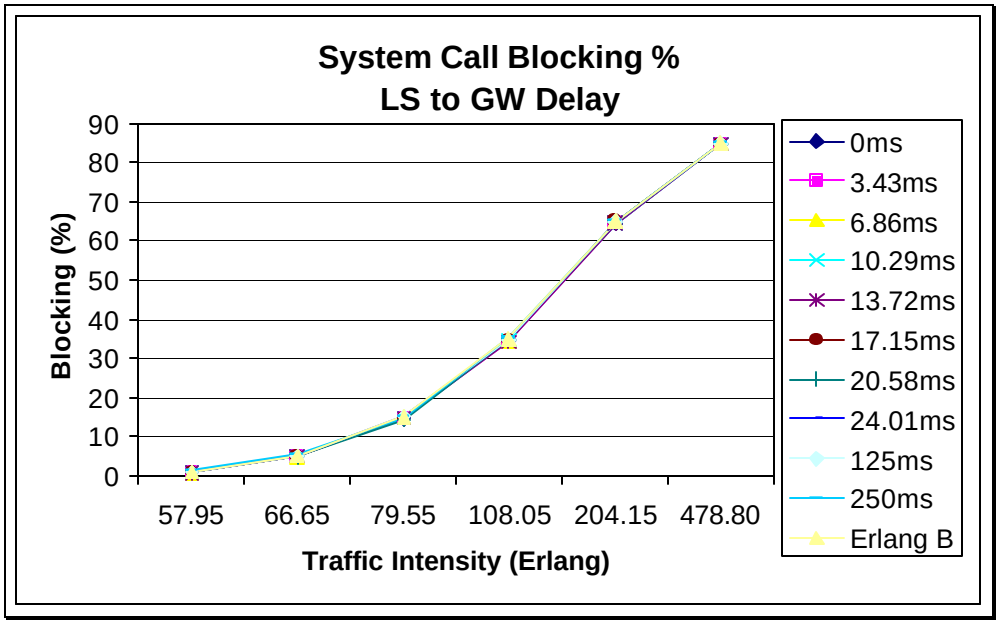

Figure 25: System Call Blocking vs. Traffic Intensity, LS-to-GW Delay Variation

Although LS-to-GW propagation delay has little impact on overall system blocking, it does have an impact on where the blocking is experienced. Figure 26 shows the predicted blocking at the LS and Figure 27 shows the predicted blocking at the GW. These results show that as the delay between the LS and GW is increased, blocking decreases at the LS and increases at the GW. At low delay, the LS curves remain close to Erlang B. At high delay, 125 ms and 250 ms, the blocking at the GW has increased significantly. The shift is caused by loss of exact knowledge of gateway dynamic resources. The increased delay is causing TRIP-lite messages to arrive at the LS late. This causes the LS to make some decisions in error. An example would be that a TRIP-lite message is sent from the GW1 to inform the LS1 it is at full trunk utilization. The correct routing decision would be to reroute the call to the LS2. The TRIP-lite message is delayed 250ms. In that time interval a new call request arrives at LS1. LS1 has not received the TRIP-lite update and does not know GW1 is at full utilization. Without that update, LS1 routes the call incorrectly to GW1. The call arrives at GW1 and it is blocked.

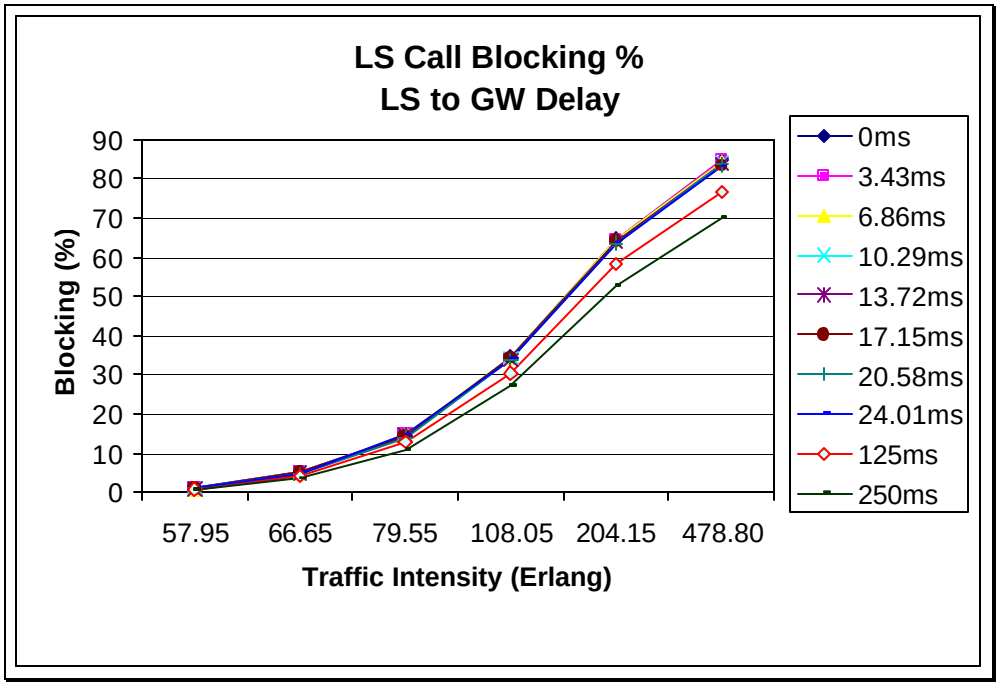

Figure 26: LS Call Blocking vs. Traffic Intensity, LS-to-GW Delay Variation

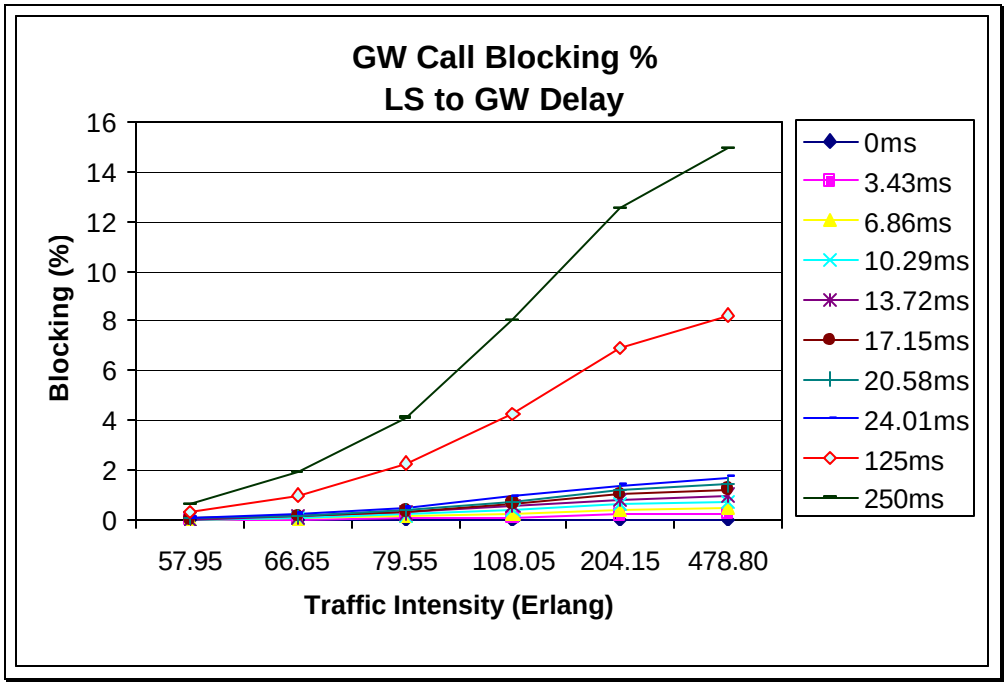

Figure 27: GW Call Blocking vs. Traffic Intensity, LS-to-GW Delay Variation

The results for LS-to-LS delay variation showed that delay had no significant impact on blocking. At low and high delay, the system remained close to predicted Erlang B values. This is as expected as the delay between the LS will not affect message updates. It only delays the rerouting of a call request to the secondary LS.

The conclusion that can be drawn is that propagation delay, and thus network topology, does not impact system blocking probability. In a TRIP-enabled network, the system blocking will be driven by traffic load and follow predicted Erlang B blocking model given the appropriate traffic assumptions. Although, as LS-to-GW delay is increased towards satellite link delays (250ms), LS knowledge of system state will degrade causing call blocking to increase at the GW. And as stated a carrier will prefer all call blocking to occur at the LS and not at the GW. The reason being that if a call is blocked at the LS, there may be opportunity for the call request to be rerouted to an alternate LS and successfully terminated.

6.2 Impact of Propagation Delay and Interarrival Rate on Call Request Rerouting between Location Servers

As described earlier the use of TRIP allows a network to react to the dynamic resources on the networks gateways. If a LS has been notified that its gateway is at full trunk utilization, that LS will look at its routing table for an alternate gateway with the same prefix destination. A model was developed to allow one reroute from the primary location server to a secondary LS. A question addressed here is how will the system reroute calls as the delay and traffic load are increased. Figure 28 shows the results of percentage of call requests rerouted versus the interarrival rate as the LS-to-GW delay is varied. Figure 29 shows the same only with LS-to-LS delay varied.

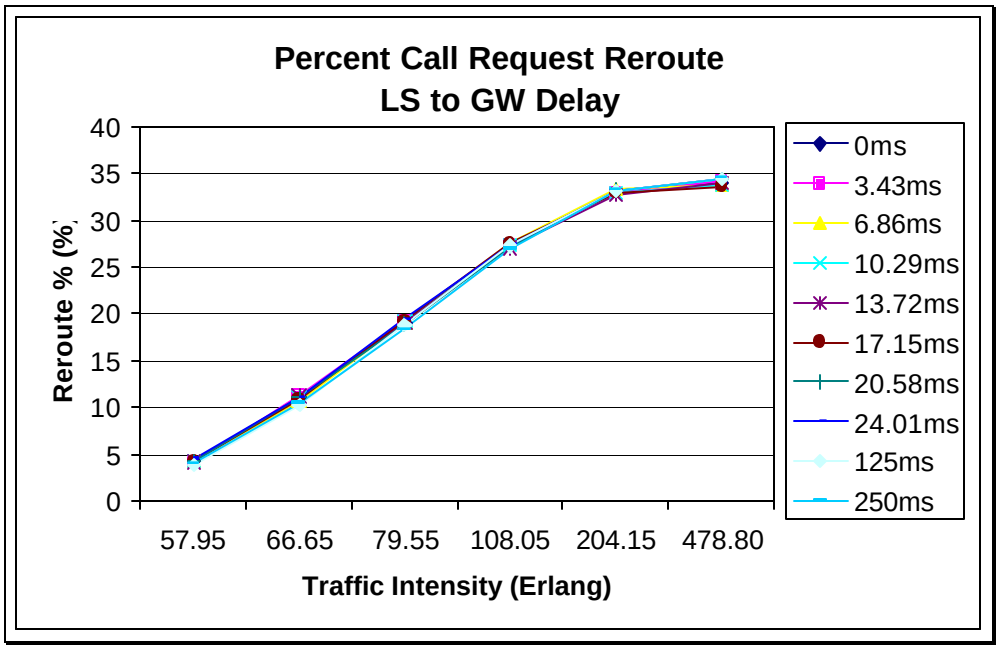

Figure 28: Percentage Calls Rerouted vs. Traffic Intensity, LS-to-GW Delay Variation

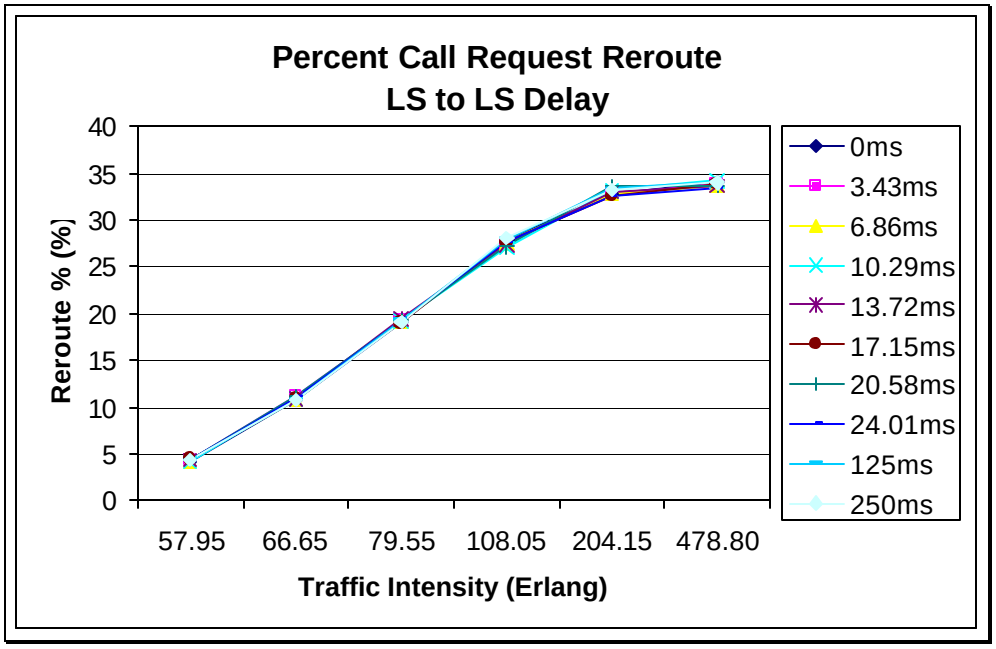

Figure 29: Percentage Calls Rerouted vs. Traffic Intensity, LS-to-LS Delay Variation

Figure 26 and 27 show that propagation delay and thus network topology does not have any impact on the percentage of call requests rerouted between location servers. The driving parameter is traffic intensity. As the load is increased, rerouting increases. At higher load, the gateways will be at full utilization more frequently forcing a higher reroute percentage.

The conclusion from this section is that network topology does not impact the percentage of reroutes in the system. The traffic intensity is the driving factor.

6.3 Impact of Propagation Delay and Interarrival Rate on Call Request Delivery to a GW

A very important characteristic of any telephony system is the amount of delay incurred due to control signaling. The TRIP model developed here addresses the evaluation of delay incurred between the generation of a call request and its delivery to a GW for termination. Additional delay would be incurred through the process of signaling between the SIP user agent and the GW and the GW signaling into the PSTN. Here delivery time to the GW is important as it shows the impact on TRIP messaging and the impacts of rerouting between location servers. Figure 30 shows the delay incurred when LS-to-GW propagation delay is increased. Every call delivered to the GW must incur the LS-to-GW delay. Thus, the LS-to-GW propagation delay is the call request delivery delay in every instance. Variation of the traffic load has no impact.

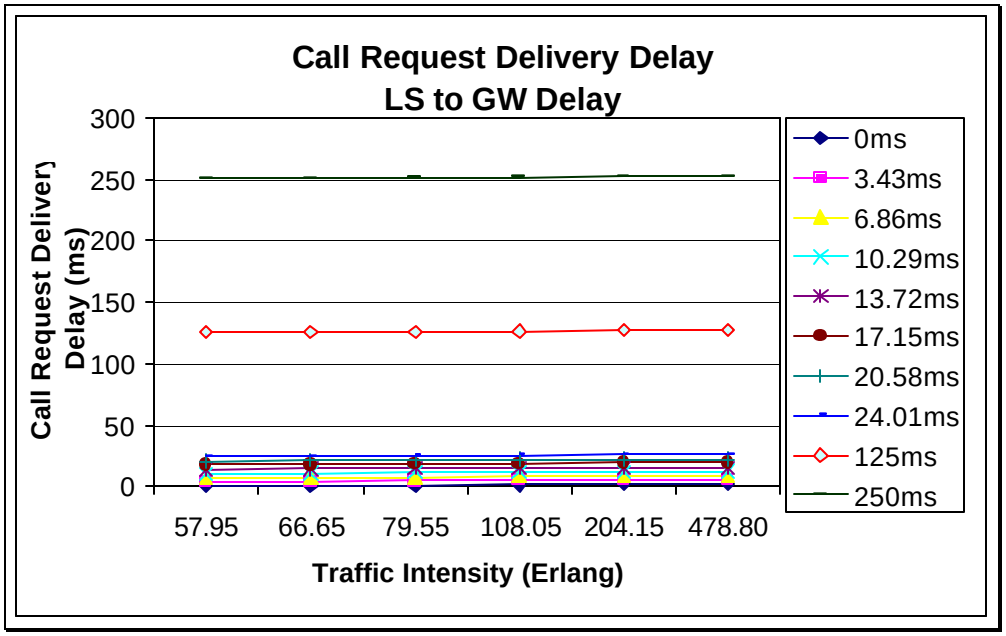

Figure 30: Call Request Delivery Delay vs. Traffic Intensity, LS-to-GW Variation

Variation of LS-to-LS propagation delay does impact call request delivery delay. Figure 29 shows the impacts. Instead of being a constant value as in the case of LS-to-GW delay, variation in the LS-to-LS delay causes increasing call request delivery delay. In this case the ability of TRIP to reroute calls is having a noticeable influence on the system. As the traffic load increases more call requests are rerouted between location servers (see Figure 28 & 29). Each rerouted call request incurs additional delay.

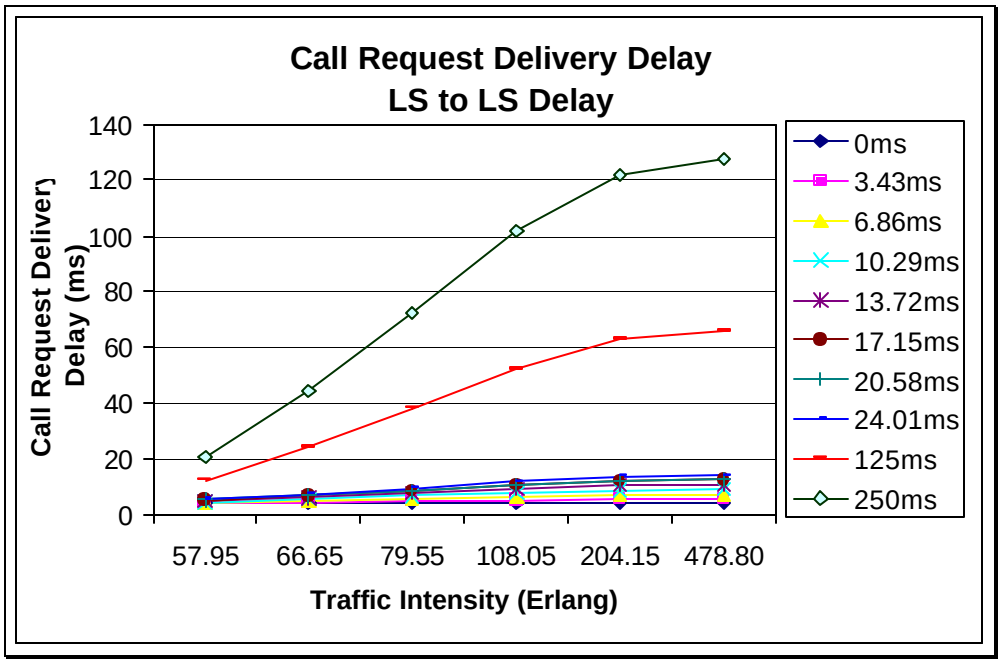

Figure 31: Call Request Delivery Delay vs. Traffic Intensity, LS-to-LS Delay Variation

A simple calculation can be done to illustrate the relationship between call request delivery delay and call request reroutes. The following calculation can be performed (as predicted in Figure 31).

- ?? At 24ms LS-to-LS propagation delay there are 4.3% calls rerouted and incur both the 24ms LS-to-LS delay plus the constant 4ms LS-to-GW delay.
- ?? Subsequently, 95.7% of call requests do not get rerouted and only incur the 4ms LS-to-GW delay.
- ?? Call Delivery Delay = 0.043*[24ms + 4ms] + 0.957*[4ms] = 5.032ms
- ?? The predicted value through simulation is 5.634ms. Thus the delivery delay can be predicted given the call request reroute percentage.

The conclusion drawn here is that LS-to-GW propagation delay will add directly to the call delivery delay. For LS-to-LS delay only a percentage of the propagation delay will add into the total call delivery delay. And that amount will be dependent upon the traffic load. As the traffic load increases, the TRIP system will be forced to reroute a higher percentage of calls between location servers, which will incur propagation delay between the location servers. The overall call delivery delay will be the sum of the LS-to-GW delay plus the LS-to-LS delay if the call was rerouted.

6.4 Comparison of a TRIP-enabled Network to a SIP Network

The main purpose of TRIP is to improve the performance of telephony routing over a SIP network. Thus, it is important to compare the performance of a TRIP network to a SIP network.

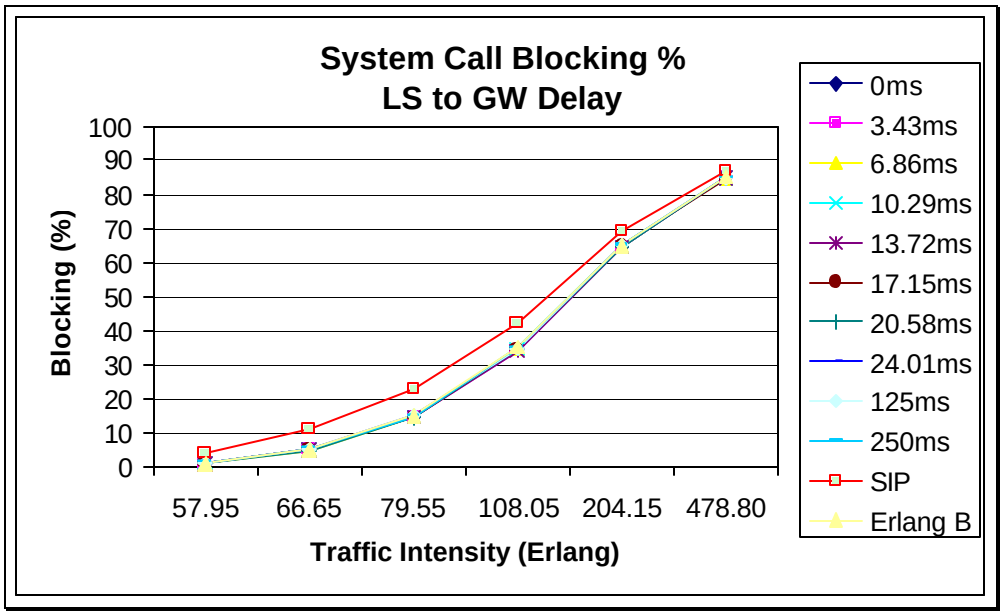

Figure 32: System Call Blocking vs. Traffic Intensity, LS-to-GW Delay Variation

Figure 32 shows the comparison of TRIP to SIP. The graph shows the TRIP delay curves at each delay, the curve predicted by Erlang B, and the simulated system

blocking curve of a SIP network. The results show that SIP blocking is consistently higher than TRIP. Additionally, it shows that given standard traffic assumptions Erlang B cannot be used to predict call blocking in a SIP network.

6.5 Impact of Trunk Failure on a TRIP Network

Also of great importance is how a TRIP-enabled network will react to a failure of trunks on a gateway. A model was built to simulate a trunk failure in GW1. At a specified time, one of the T1s was removed from service. This dropped the available trunks on GW1 from 48 to 24. GW2 would still have 24. The overall system capacity dropped from 72 to 48 trunks. The performance issue is how the system reacts to the lost trunks and the subsequent restoral of the lost trunks. Initially, the model looked at call blocking versus time but the results were influenced by both the impact of variance in the estimator for blocking probability and system dynamics. To mitigate the impact of variance in the estimator for blocking probability, the model was used to plot system blocked calls versus time instead of call blocking. The results show the reaction to the new system state when trunks are lost and after trunks are restored. Figure 33 shows number of blocked calls versus time for a 1% blocking system with varied LS-to-LS propagation delay.

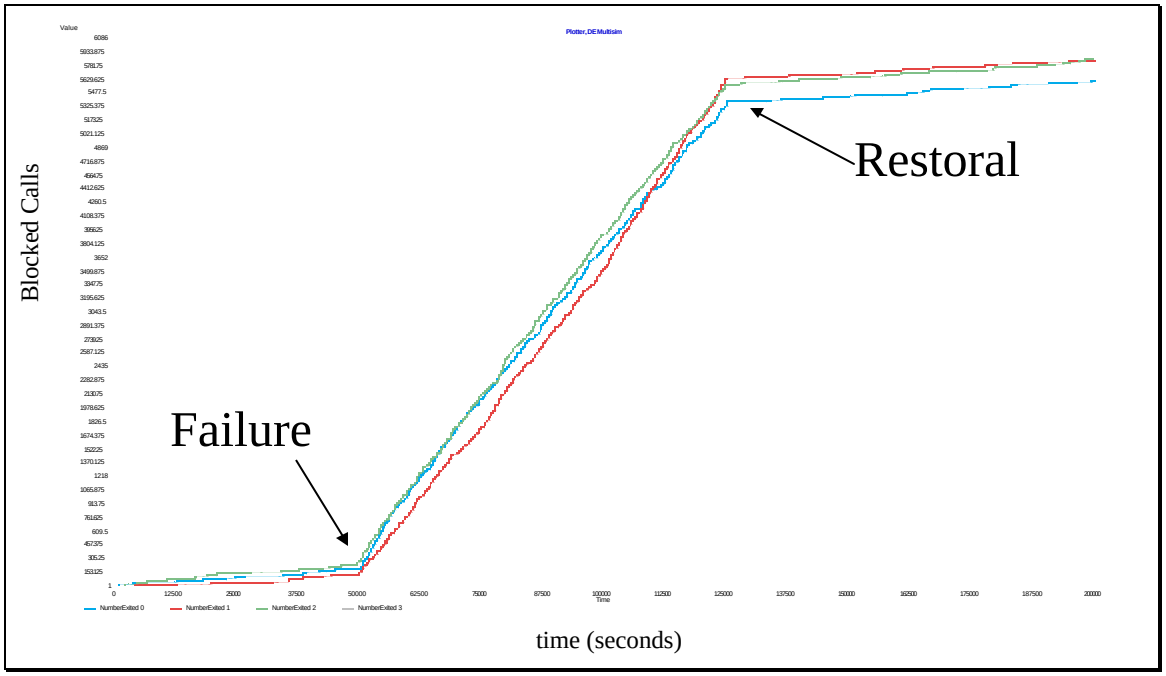

Figure 33: Cumulative number of blocked calls vs time, 1% call blocking, LS-to-LS Delay Variation, Blue: 0ms; Red: 24ms; Green: 250ms

Figure 33 shows the results for each of the three delay values (0ms, 24ms, and 250ms). The figure shows propagation delay has little impact on results. The slopes of each interval can be used to show the system reacts as Erlang B would predict. The simple calculation below can be used to illustrate the relationship between traffic load and the slope during each interval. Ten runs were executed and an average slope was calculated for each interval, before failure, after trunk failure and after trunk restoral.

?? Prior to trunk failure the average slope is 0.00333 blocked calls/second.
?? The traffic intensity is [57.95 Erlang] / [180second] = 0.3219 Erlang/second.
?? Call Blocking Percentage = [slope / traffic intensity] * 100 = 1.03%
?? The Erlang B predicted value for a system with 72 trunks and 57.95 Erlang is 1% call blocking. Thus the system is performing as expected prior to trunk failure.

?? After trunk failure the average slope is 0.07254 blocked calls/second.
?? The traffic intensity is [57.95 Erlang] / [180second] = 0.3219 Erlang/second.
?? Call Blocking Percentage = [slope / traffic intensity] * 100 = 22.5%
?? The Erlang B predicted value for a system with 48 trunks and 57.95 Erlang is 22.1% call blocking. Thus the system is performing as expected during trunk failure.

?? After trunk restoral the average slope is 0.00348 blocked calls/second.
?? The traffic intensity is [57.95 Erlang] / [180second] = 0.3219 Erlang/second.
?? Call Blocking Percentage = [slope / traffic intensity] * 100 = 1.08%
?? As before the Erlang B predicted value for a system with 72 trunks and 57.95 Erlang is 1% call blocking. Thus the system is performing as expected after trunk restoral.

Table 1 shows the results of varying LS-to-LS and LS-to-GW propagation delay.

Delay Location	**Delay (ms)**	**Interval**	**Average Slope (blocked/sec)**	**Calculated Call Blocking (%)**	**Erlang B (%)**
LS-to-LS	0ms	Before Failure	0.00328	1.02%	1.0%
LS-to-LS	0ms	After Failure	0.07155	22.2%	22.1%
LS-to-LS	0ms	After Restoral	0.00312	0.97%	1.0%
LS-to-LS	24ms	Before Failure	0.00333	1.03%	1.0%
LS-to-LS	24ms	After Failure	0.07254	22.5%	22.1%
LS-to-LS	24ms	After Restoral	0.00348	1.08%	1.0%
LS-to-LS	250ms	Before Failure	0.00315	0.97%	1.0%
LS-to-LS	250ms	After Failure	0.07203	22.3%	22.1%
LS-to-LS	250ms	After Restoral	0.00342	1.06%	1.0%
LS-to-GW	0ms	Before Failure	0.00347	1.08%	1.0%
LS-to-GW	0ms	After Failure	0.07013	21.8%	22.1%
LS-to-GW	0ms	After Restoral	0.00378	1.2%	1.0%
LS-to-GW	24ms	Before Failure	0.00331	1.03%	1.0%
LS-to-GW	24ms	After Failure	0.07181	22.3%	22.1%
LS-to-GW	24ms	After Restoral	0.00344	1.07%	1.0%
LS-to-GW	250ms	Before Failure	0.00352	1.09%	1.0%
LS-to-GW	250ms	After Failure	0.07217	22.4%	22.1%
LS-to-GW	250ms	After Restoral	0.00327	1.02%	1.0%

Table 1: TRIP Results During Trunk Failure with Varied Propagation Delay

As shown in Table 1, the analysis was performed at each delay, (0ms, 24ms, and 250ms), for both LS-to-LS propagation delay and LS-to-GW propagation delay. The

results show propagation delay does not impact system reaction to a state change such as gateway trunk failure. The simulated call blocking results at each propagation delay value are nearly identical to predicted Erlang B values.

Additionally, the results show the system reacts within a reasonable time after state change is introduced. When the failure is initiated, Figure 33 shows the slope of the curve increased just after the state change. Also, this abrupt reaction is seen after trunk restoral. Figure 34 shows a magnified view of the results just after trunk restoral. It shows that the system blocking slope just following restoral tends to zero just after state transition. The state change increased the available trunks which causes the near zero call blocking.

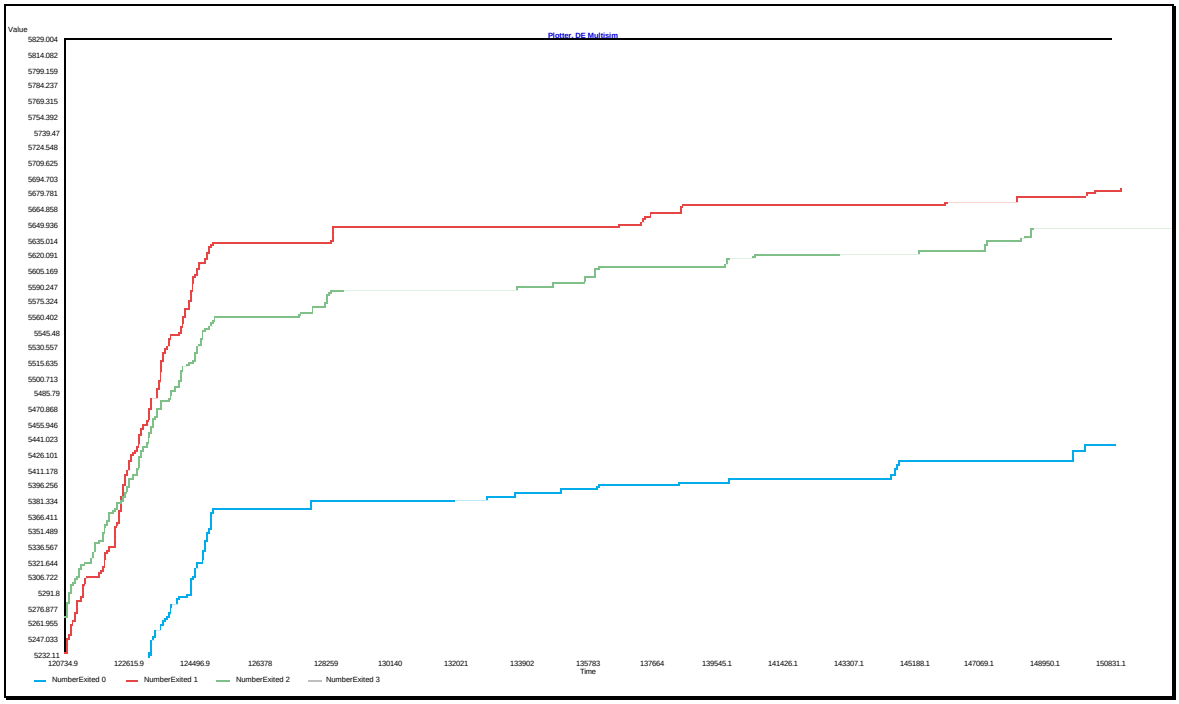

Figure 34: Magnified View just after Trunk Restoral: System Blocked Calls vs. Time, 1% call blocking, LS-to-LS Delay Variation, Blue: 0ms; Red: 24ms; Green: 250ms

The results show that the system reacts within a reasonable time interval to new state. When a trunk failure is introduced the system tends to the new call blocking value predicted by Erlang B and also reacts appropriately when the trunk is returned to service. Additionally, the results are impacted by the relative time between the element-to-element propagation delay and the call request interarrival time. The interarrival time between call requests is longer than propagation delay. Thus, a small number of call requests (approximately 1-3) will have arrived prior to delayed TRIP-lite messaging arriving at the location server. For 1% call blocking, the TRIP system will react to the state change in approximately 3.1-9.3 seconds, which is equivalent to 1-3 call request arrivals. As the traffic load is increased, the system reaction time to the state change will decrease. This is a result of call requests arriving at a faster rate. For a 15% call blocking system, the restoral time is approximately 2.3-6.9 seconds. For an 85% call clocking system, the restoral time is approximately 0.38-1.14 seconds.

The conclusion is the time required for a TRIP system to react to a change in state is based on traffic load. As the traffic load is increased, the system reaction time to the state change will decrease. Additionally, the results show that propagation delay during a failure scenario does not impact the system reaction to new state.

6.6 Confidence Interval of TRIP Simulation

The use of error bounds will provide confidence in the simulation results presented here. The analysis in [15] will be used to locate a confidence interval for the simulated results. The confidence interval will be given by the following probability expression from [15].

$$P\left\{ p \le \frac{N}{N + d_{\gamma}^2}\left[\hat{p} + \frac{d_{\gamma}^2}{2N} + d_{\gamma}\left[\frac{\hat{p}(1-\hat{p})}{N} + \frac{d_{\gamma}^2}{4N^2}\right]^{1/2}\right] \right\} = 1 - \gamma$$

where: $1 - \gamma$ = confidence coefficient

N = total number of call requests which is a function of interarrival rate

d_{γ} = normal variate of the desired confidence

$\hat{p}$ = sample mean of the simulated blocking value.

p = actual value.

The simulations run for 1% blocking had the fewest calls generated. Thus, all other simulation results would have tighter confidence intervals as compared to this case. The number of calls (events) simulated in each run multiplied by 10 runs gives total events for the set of simulations. From [15] the number of events simulated for this case provides a confidence coefficient of 1-? ?? ???? ???

$1 - \gamma = 99\%$

$N_{1\%}$ = [57.95 Erlang / 180 seconds] * 30,000 seconds = 9,658 calls per run

$N_{1\%Total}$ = [10 runs] * 9,658 = 96,580 calls (events).

$d_{\gamma} = 2.576$ ($Q(d_{\gamma}) = 0.005$).

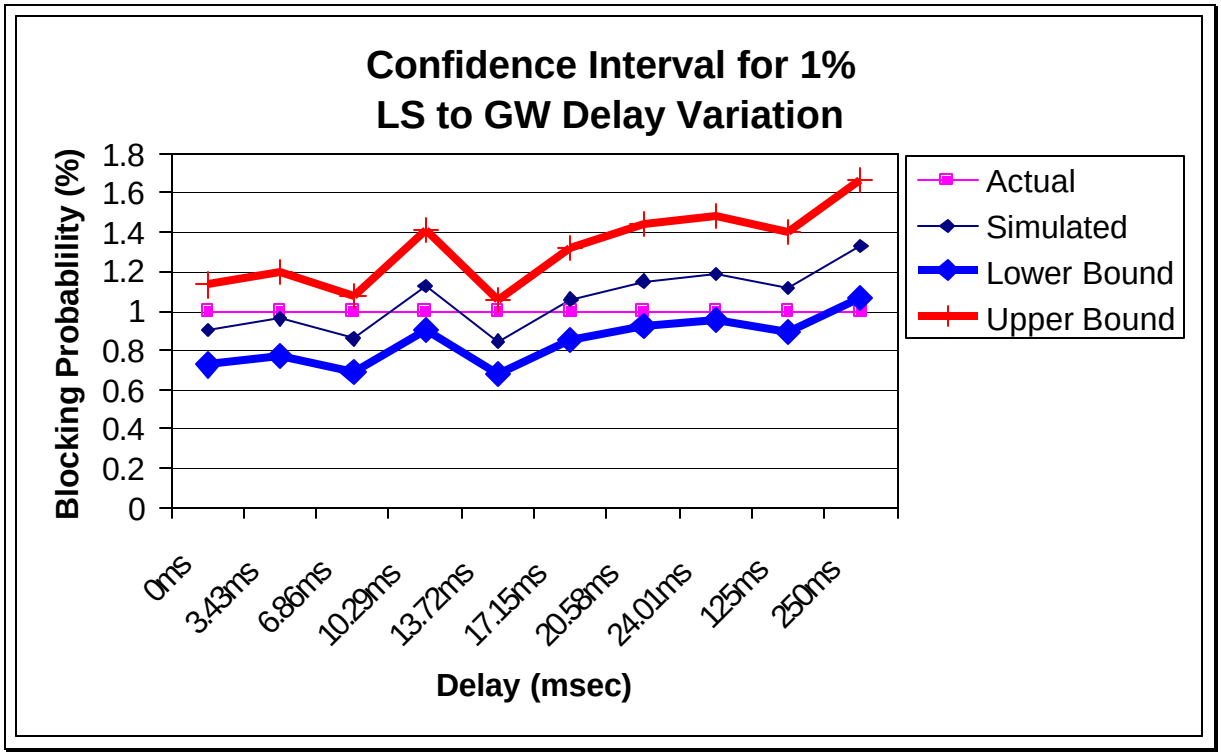

Figure 35: Confidence Interval for 1% System Blocking

Figure 35 shows the plot of the confidence intervals around the simulated blocking probability. It shows that all simulated results lie inside the confidence bounds. Therefore, we have a high confidence in the results presented here.

7.0 Summary of the Performance Evaluation of TRIP

Throughout Section 6, conclusions have been drawn based on results derived from the TRIP simulation model. Below are all conclusions and discussion about the impact of each case studied.

?? Network topology does not impact system blocking probability. In a TRIP-enabled network, the system blocking will be driven by traffic load.
   - This result impacts geographic deployment of location servers to support the network. From a system blocking standpoint, designers do not need to be concerned with propagation delay but must be concerned with traffic load.

?? Overall system blocking will follow Erlang B given specified values for traffic load and call holding time.
   - This result allows designers to implement a correctly sized TRIP network based on forecasted customer usage. This would impact number of trunks to support a given destination prefix, number of gateways in a geographic area, and number location servers in the network.

?? As Location Server-to-gateway (LS-to-GW) delay is increased towards a satellite link delay (250ms), loss of knowledge about the current state of the system causes call blocking to increase at the GW. A carrier will prefer all call blocking to occur at the LS and not at the GW. The reason being that if a call is blocked at the LS, there may be opportunity for the call request to be rerouted to an alternate LS and successfully terminated.
   - This result places a limit on implementation options. TRIP messaging can incur propagation delay equivalent to cross country fiber links but satellite links should not be considered.

?? Propagation delay, LS-to-GW and Location Server to Location Server (LS-to-LS), does not impact the percentage of reroutes in the system. The traffic intensity is the driving factor.
   - This result dictates that designers be concerned with traffic load and not propagation delay when addressing TRIP rerouting functionality.

?? LS-to-GW propagation delay will add directly to the call delivery delay. For LS-to-LS delay only a percentage of the propagation delay will add into the total call delivery delay. And that amount will be dependent upon the traffic load. As the traffic load increases, the TRIP system will be forced to reroute a higher percentage of calls between location servers, which will incur propagation delay introduced between the location servers.
   - This issue impacts the delay budget. The result indicates that any delay between the LS and GW must be added to overall call setup delay. While, only a percentage of the delay between LS and LS should be added. And that the delay addition is dependent upon rerouting and traffic load.

?? SIP blocking is consistently higher than TRIP and higher than what would be predicted by Erlang B. This shows that a TRIP-enabled network can achieve better performance than a SIP network cannot.

- This is a very important result in that TRIP provides a SIP network with lower blocking. It benefits the carrier with less provisioning, gateway dynamic resource information available at the proxy, optimum path routing, and also better blocking performance.

?? The time required for a TRIP system to react to a change in state (i.e., gateway trunk failure) is based on traffic load. As the traffic load is increased, the system reaction time to the state change will decrease. Additionally, the results show that propagation delay during a failure scenario does not impact the system reaction to new state.

- Network failures occur. This result shows that when a failure happens the TRIP network will react within a reasonable time interval and tend toward the new steady state.

8.0 Next Steps

This document has provided a detailed understanding of a new signaling protocol being developed to support voice telephony routing. The protocol is Telephony Routing over IP (TRIP). The most basic function of TRIP is to locate the optimum gateway out of a Voice over IP (VoIP) network into the Public Switched Telephone Network (PSTN) [9]. This document has included a background section on signaling protocols, including TRIP, a TRIP simulation model test plan, a description of the TRIP simulation model, simulation results, and conclusions.

This section will provide additional areas of investigation beyond this thesis and the simulation model.

- ?? Simulation of TRIP-lite network with added update messaging. As stated in Section 5.2, the model developed here updates the LS only when the gateway is at full trunk utilization.
    - o Evaluation of a TRIP system with added messaging would determine what performance impacts the added LS knowledge would provide the network.
- ?? Simulation of TRIP network synchronization (TRIP Routing Convergence).
    - o TRIP is a routing protocol and like other routing protocols convergence time is crucial to performance. Network designers must understand TRIP convergence intervals to know how a real TRIP network will perform.
- ?? Lab evaluation of Vendor TRIP-lite equipment and software.
    - o Evaluation of vendor TRIP-lite equipment will validate simulation results. Additionally, it is important to know if a vendor's TRIP-lite solution performs appropriately.
- ?? Lab evaluation of Vendor Interior Administrative Domain Routing (I-TRIP) equipment and software.
    - o Evaluation of vendor I-TRIP equipment will validate simulation results. Additionally, it is important to know if a vendor's I-TRIP solution performs appropriately.
- ?? Lab evaluation of Vendor Exterior Administrative Domain Routing (E-TRIP) equipment and software.
    - o Evaluation of vendor E-TRIP equipment will validate simulation results. Additionally, it is important to know if a vendor's E-TRIP solution performs appropriately.
- ?? Lab evaluation of a TRIP network with all TRIP entities.
    - o A full evaluation with each TRIP entity present will provide an understanding of full network performance.

After each of the above steps are investigated, network designers will be prepared to implement TRIP into a SIP network supporting customers. The simulation and lab evaluation will provide tangible design characteristics, which will support design and implementation.

9.0 Bibliography

1. B. Goode, “Voice over Internet Protocol (VoIP),” Proceedings of IEEE, Volume 90, Number 9, Pages 1495-1517, September 2002.

2. A.R. Modarressi and A. Skoog, “Signaling System No. 7: A Tutorial,” IEEE Communications Magazine, pages 19-35, July 1990.

3. International Engineering Consortium, “SS7 Signaling endpoints in a Switched-Circuit Network,” http://www.iec.org/online/tutorials/ip_in/topic01.html, 2002.

4. G. Willmann and P. Kuhn, “Performance Modeling of Signaling System No. 7,” IEEE Communications Magazine, pages 44-45, July 1990.

5. K. Morneault, S. Rengasami, et al., “ISDN Q.921-User Adaptation Layer”, Request for Information #3057, Pages 1-70, February 2001.

6. L. Ong, I. Rytina, et al., “Framework Architecture for Signaling Transport,” Request for Information #2719, pages 1-24, October 1999.

7. A. Neogi and T. Chiueh, “Performance Analysis of an RSVP-Capable Router,” IEEE Communications Magazine, pages 56-57, October 1999.

8. R. Braden, L. Zhang, et al., “Resource ReSerVation Protocol (RSVP), Version 1 Functional Specification,” Request for Information #2205, Pages 1-36, September 1997.

9. H. Schulzrinne and J. Rosenberg, “The Session Initiation Protocol: Internet Centric Signaling,” IEEE Communications Magazine, Pages 134-140, October 2000.

10. P. Zimmerman, D. Atkins, et al., “PGP Message Exchange Formats,” Request for Comments #1991, Pages 1-21, August 1996.

11. H. Sinnreich, “Internet Communications Enabled by SIP,” http://www.sipforum.org/whitepapers, Page 2-4, August 2000.

12. J. Rosenberg, H. Salama, and M. Squire, “Telephony Routing over IP (TRIP),” Request for Comments #3219, Pages 1-22, January 2002

13. H.323 Forum, http://www.h323forum.org, 2002.

14. Extend Model Software & Documentation, http://www.imaginethatinc.com, 2002.

15. M. Jeruchim, P. Balaban, & K. Shanmugan, “Simulation of Communication Systems,” Pages 496-501, 1992.