Characterizing Session Initiation Protocol (SIP) Network Performance and Reliability

Vijay K. Gurbani, Lalita J. Jagadeesan, and Veena B. Mendiratta

Bell Laboratories, Lucent Technologies
Naperville, Illinois
{vkg,lalita,veena}@lucent.com

Abstract. The Session Initiation Protocol (SIP) has emerged as the preferred Internet telephony signaling protocol for communications networks. In this capacity, it becomes increasingly essential to characterize both the performance and the reliability of the signaling entities utilizing the protocol. We provide an analytical look at the performance of a SIP network as well as a reliability model of SIP servers. Keywords: SIP, Stochastic Processes, Queueing Analysis, Performance Analysis, Reliability Analysis.

1 Introduction

The Public Switched Telephone Network (PSTN) has evolved over a century to become an integral part of human communications. Over the years, the network has been tuned for performance and has evolved to become highly reliable, with individual switches experiencing only a few seconds of downtime per year. As the telecommunications industry moves towards a new network (the Internet) with a new set of signaling protocols, media behaviors and routing protocols – which are markedly different from the PSTN model – is it reasonable to assume that the performance and reliability metrics established in the PSTN are applicable and achievable in the new environment?

Performance analysis and reliability of circuit-based communication networks has been well studied. Models exist in the PSTN that characterize performance in telecommunication switches. One measure of performance in the PSTN is the Busy Hour Call Attempt (BHCA) metric, which is defined as the number of call attempts during the busiest hour of the day. The BHCA measures the capacity of a PSTN call processing switch in terms of the total number of calls arriving at a switch during peak periods. In commercial PSTN switches, it ranges from 1 million to 2 million calls per hour. Another measure of performance is the switch cross-office delay, where a typical value is 100-300 milliseconds (ms); the precise requirements for this metric are specified by signaling message type.

Circuit switches for voice meet stringent requirements for reliability with expected switch availability greater than 0.99999 and expected call loss of the order of tens per million calls handled. For call loss, in the event of failures, the priority is to save calls in progress over calls in the setup stage. This high reliability is achieved through

M. Malek, E. Nette, and N. Suri (Eds.): ISAS 2005, LNCS 3694, pp. 196 – 211, 2005.

redundancy of the switch elements, robust software and the implementation of hardware and software fault tolerance mechanisms at various layers in the system.

Current trends in the telecommunications industry favor voice over Internet Protocol (VoIP) technology. The introduction of the Session Initiation Protocol (SIP) and the widespread adoption of the protocol by both wireless and wireline telecommunication players has accelerated the trend. If VoIP is to become the pervasive telecommunication model, then the performance and reliability of call processing elements in the Internet needs to be on par with those of the circuit-switched elements. To this end, there are three contributions of this paper. The first is to provide analytical models for the performance analysis of a SIP network and use the models to analyze the performance of a SIP network with respect to varying arrival rates, service rates and network delays. The network delay is characterized using one intermediary as well as a chain of intermediaries of varying length. The second contribution of the paper is evaluating a SIP network for reliability and lost calls. Given the industry trend towards using commercial-off-the-shelf hardware and software components, our evaluation is based on utilizing generally available application layer fault tolerance mechanisms as opposed to using proprietary solutions implemented at lower layers. Finally, we compare our findings with the established norms of PSTN performance and reliability.

The rest of the paper is organized as follows: Section II covers existing work related to SIP performance. Section III provides a brief background on the mechanisms of signaling exchange in the PSTN and a SIP network. Section IV details the performance model and the results from the performance analysis. Section V presents a reliability model combined with the performance model and the subsequent results. We conclude the paper by summarizing our observations and future work to be done in this area.

2 Related Work

Wu et al. [2] analyze SIP performance in light of SIP-T (SIP for Telephones) [3]. SIP-T is an effort to provide the integration of legacy telephone signaling into SIP messages through encapsulation and translation. The PSTN call setup messages that would normally flow between two PSTN switches are encapsulated and transported as a payload over a SIP network connecting two PSTN islands. SIP-T also translates certain PSTN call setup headers into their closest SIP equivalent to enable intermediaries to route the request. Wu et al. analyze the queuing delay and queuing delay variation using embedded Markov chains in a M/G/1 queuing model. Our work, by contrast, analyzes performance under varying arrival rates, service rates and network delays of an end-to-end native SIP ecosystem which includes multiple intermediaries (SIP proxies). We also analyze the reliability, including call loss, of SIP signaling entities through a hierarchical performance and reliability model.

The SIPStone benchmark [4] is an early attempt at characterizing server performance in a way that is useful for dimensioning and provisioning a SIP network. One of the aims of SIPStone is to enunciate a repeatable set of experiments in order to compare different implementations across the uniform set of experiments. It assumes the standard SIP trapezoid: a client conversing with a SIP intermediary, which in turns converses with a destination server. Our work builds in part on SIPStone to provide an analytical view of performance and reliability across a wider spectrum which includes modeling a SIP network using one intermediary, and a chain of intermediaries.

Zhu [10] analyzes the usage of SIP in the Third Generation Partnership Project's (3GPP) IP Multimedia Subsystem (IMS). This analysis involves the usage of SIP in the context of a centrally controlled architecture, which imposes additional requirements on the protocol above and beyond those specified in [1]. Our analysis is based on the protocol as specified in [1].

Lipson [12] presents an approach for using model checking of Markov Reward Models to analyze properties of a simple SIP network. The focus is on transient properties related to the number of calls processed prior to system failure or system repair. Rewards are expressed as simple rates of incoming requests for call setups. Our model, in contrast, is a hierarchical model consisting of a high-level Markov Reward Model and a lower-level queuing network model. Furthermore, our model considers implications of different fault tolerance approaches and we use closed-form equations rather than model checking to analyze properties of our model.

3 Background

In order to study the performance of telecommunications systems, it is instructive to understand the entities involved in call setup. We provide a brief overview of call setup in the PSTN and compare it with call setup in the Internet using SIP.

PSTN Call Setup. In the PSTN, telephone users connect through the telephone system into the central office (CO). Hundreds of COs may be installed in a metropolitan area. Telephone traffic from end users terminates at the CO through a pair of wires (or four wires) called the local loop or the subscriber loop. Telephone traffic from the COs is generally aggregated into trunks and carried to a toll/tandem office from where it is distributed to other toll offices. High usage trunks are established when the volume of calls warrants the installation of high capacity between two offices.

A salient point about the PSTN is that the network used to route the media stream between switches is different from the network used to route signaling messages. Signaling messages between switches are routed over a packet-based network called Signaling System Number 7 (SS7). Communicating switches exchange SS7 messages to setup a call by allocating media resource end-to-end. Once the media resources have been allocated and the call has been set up, the voice flows over direct media connections between each intervening switch. More information about PSTN signaling is available in [7].

Call Setup in SIP. SIP [1] is an application-layer protocol used to establish, maintain and tear down multimedia sessions. It is a text-based protocol with a request-response paradigm. A SIP ecosystem consists of user agents, proxy servers, redirect servers, and registrars. Of special interest to us with respect to this paper are user agents and proxy servers.

There are two types of SIP user agents: a user agent client (UAC) and a user agent server (UAS). A UAC and a UAS are software programs that execute on a computer, an

Internet phone, or a personal digital assistant (PDA). A UAC originates requests (i.e. start a phone call) and a UAS accepts and acts upon a request. UASes typically register themselves with a registrar, which binds their current Internet Protocol (IP) address to an email-like identifier used to identify the user. This registration information is used by SIP proxy servers to route the request to an appropriate UAS.

Proxy servers are SIP intermediaries that provide critical services such as routing, authentication, and forking. A SIP proxy, upon the receipt of an incoming call setup request, will determine how to best route the request to a downstream UAS.

The request to establish a session in SIP is called an INVITE. An INVITE request generates one or more responses. Responses to requests indicate success or failure, distinguished by a status code. Responses with status code 1xx (100-199) are termed provisional responses and serve to update the progress of the call; the 2xx code is for success and higher number for failures. 2xx-6xx responses are termed as final responses and serve to complete the INVITE request. The INVITE request is forwarded by a proxy (through possibly another chain of proxies) until it gets to its destination. The destination sends one or more provisional responses followed by exactly one final response. The responses traverse, in reverse order, over the same proxy chain as the request. Figure 1 provides a time-line of call establishment between a UAC and a UAS. The request is forwarded through a chain of proxies.

With reference to Figure 1, the UAC sends an INVITE to P1 and P1 routes the call further downstream. From the UAC's reference, P1 is called an outbound proxy. P1 determined that the request should be forwarded to P2 (the UAS is in a different domain). When the request arrives at P2, it queries its location server and further proxies the request to the UAS. From the UAS point of view, P2 is the inbound proxy. The UAS issues a provisional response followed by a final response. The call is setup when the UAC receives the final response.

Comparing SIP entities to the PSTN, the UAS and UAC correspond to phones; proxies act as 'switches'. However, unlike the PSTN, there is no signaling overlay network. Both media and call signaling use the same network. Nor is there a notion of a toll/tandem switch in the Internet. The routing fabric of the Internet assures that packets containing voice or data are forwarded to their intended destination. More information on Internet telephony signaling and SIP is available in [1, 8, 9].

4 Performance Analysis

The performance measures of interest for SIP networks are the steady-state mean response time and mean number of jobs in system. The mean response time of a proxy server is defined as the mean elapsed time from the time t_1 an INVITE request from an User Agent Client (UAC) arrives at the proxy server until the time t_2 that the proxy server sends a final response to the UAC. The mean number of jobs in system is defined as the mean number of calls currently being set up or waiting to be set up by the proxy server. Also of interest is the behavior of these performance measures as a function of the mean arrival rate of incoming INVITE requests, the mean service rates for processing SIP requests/responses, and the mean propagation delay between adjacent SIP proxy servers in the network.

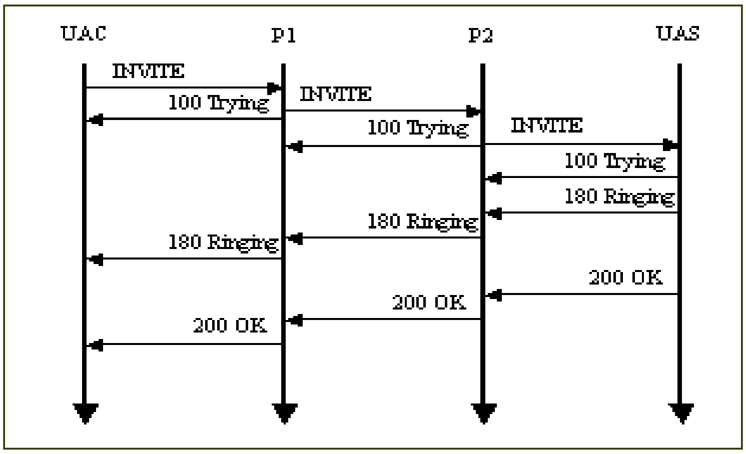


Fig. 1. SIP call establishment

4.1 Performance Model and Assumptions

We model a SIP proxy server as an open feed-forward queuing network, in which arriving jobs correspond to INVITE requests received by the SIP proxy server from an upstream UAC in a SIP network. The queuing network consists of sequences of queuing stations that correspond to possible sequences of SIP requests and responses during a call setup. Each queuing station does the servicing of the SIP request/response at the corresponding point in the call setup sequence. In constructing our model, we made some simplifying assumptions. First we model a "180 Ringing" response and assume that immediately following this will be a final response (either a 2xx final response or non-2xx final response). When an INVITE request arrives at the proxy, it is sent downstream and may engender a "180 Ringing" response or a non-2xx final response.

Next, we make certain assumptions about the mean service time. In SIP, mean service time will vary by implementation. For this analysis, we assume that it takes $1/\mu$ mean time to service an INVITE request at a proxy and derive other service time parameters from this base service time. Servicing a SIP message includes extracting the message from the transport layer, parsing it, performing a location server lookup, querying the DNS and serializing the request on a connection opened with the next downstream entity. In response to the INVITE, the proxy will receive a 180, a 200, or a non-200 response. Since the effort required to process a response is far less than that for processing an INVITE, we assign a mean service time of $0.3/\mu$ for processing 180, 2xx and non-2xx responses.

For simplicity, we assume a lossless network. This is not an unreasonable assumption, loss rates of 10^{-7} are not uncommon in Internet2 [13]. Operational networks will typically have very low packet loss rates to maintain good voice quality and acceptable call setup delays. Finally, we assume a simple call flow from a UAC to an outbound proxy, which transmits the call to an inbound proxy in the domain of the UAS and from there it arrives at the UAS. For this call flow, we model two cases: one, the inbound proxy is the same as the outbound proxy (UAC ⇒ P ⇒ UAS), and, two, there is a chain of proxies between the UAC and the UAS (UAC ⇒ P_1 ⇒ P_2 ⇒ … ⇒ P_N ⇒ UAS). We do not consider advanced SIP services such as forking. Figure 2 shows the basic model.

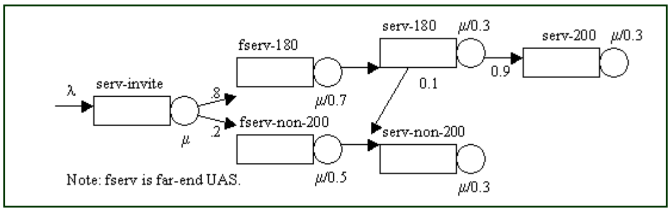


Fig. 2. Model with no network delays

When an INVITE arrives at a proxy, with a probability of 0.8 it will engender a "180 Ringing" response, and with a probability of 0.2 it will result in a failure response. The failure leg models the behavior of a call that was not setup. Following the model further, we note that with a probability of 0.9, the "180 Ringing" is followed by a "200 OK" response; i.e. the user associated with the UAS successfully answered the call. With a probability of 0.1, we model the "180 Ringing" resulting in a non-2xx response; i.e. the UAS was successfully contacted but the user did not pick up the phone. The fserv-180 and fserv-non-200 stations model a UAS. A UAS does not proxy a request downstream; instead, it issues a response. As such, it requires less computation than what a proxy undergoes when it services a request. Hence, in the model, we have assumed a mean service time of $0.7/\mu$ for sending the 180 followed by a 200 or non-2xx response. Similarly, sending only a non-2xx response takes even less time, modeled by a mean service time of $0.5/\mu$. Note that we assume that there is zero delay between the "180 Ringing" response and the "200 OK" response. In real systems there will be a variable delay — this is the time taken by the user to answer the call. The length of this delay interval would impact the number of jobs in system performance measure and also has implications for checkpointing.

In the base queueing model of a SIP proxy server depicted in Figure 2, each queueing station is modeled as a M/M/1 queue. This model is an open, feed-forward queueing network, since jobs arrive from an outside source, and there is no feedback among queueing stations in the queueing network. Using standard approaches [11], the mean number of jobs N in system is given by $N = \sum^{J}_{k=1} \rho_k/(1 - \rho_k)$, where $\rho_k = \lambda_k/\mu_k$, $\lambda_1 = \lambda$, $\lambda_j = \sum_{k=1}^{j-1} (\lambda_k Q[k,j])$ for $1 < j \le J$, and J=6 is the number of stations in the queuing model. Q is the one-step probability matrix corresponding to the queuing model, that is, Q[*i*,*j*] is the probability that a job departing station *i* goes to station *j*. Since the queuing network is feed-forward, we assume that the serv-INVITE station corresponds to station 1, and the other stations are numerically ordered in the above equations so that Q[i,j] = 0 for all i ≥ j. The mean response time *R* for jobs is then given by Little's law [11], R = N/λ.

We now extend this model to include propagation delays between adjacent SIP proxy servers and UAC and UAS's in call setup paths**.** Propagation delays can be modeled through a delay server; namely, a M/M/∞ queuing station with mean service time given by the mean propagation delay. The extended model is shown in Figure 3.

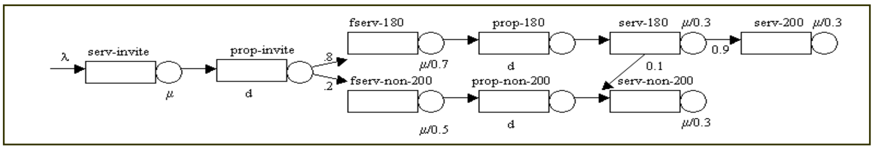


Fig. 3. Model with network delay

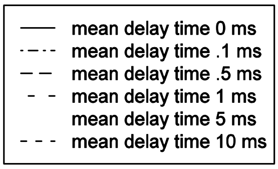
The prop-INVITE station models the propagation delay in proxying the INVITE request to the downstream SIP entity, while the prop-180 station models the propagation delay in receiving a 180 response, together with a 200 response or non-2xx response, from the downstream SIP entity. The prop-non-200 station is similar.

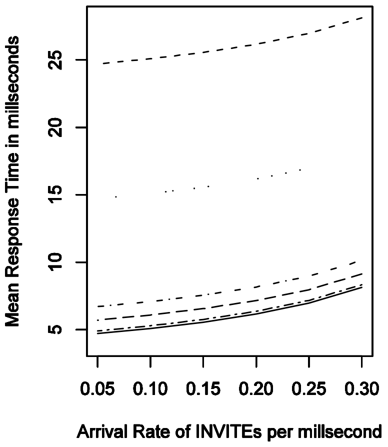
The mean number of jobs in M/M/∞ stations is given by the arrival rate of jobs into the station multiplied by the mean service time (i.e. mean propagation delay in our model) [11]. It is thus straightforward to extend the earlier equations to compute mean response time and mean number of jobs in system for this extended model. Note that the model in Figure 2 corresponds to the extended model with propagation delay of zero.

4.2 Results of Performance Analysis for SIP Proxy Servers

Using this approach, the mean response time for a proxy server is computed; the results are shown in Figure 4. The plots show propagation delays from 0 to 10 ms, corresponding to distances of 0 to 1000 miles between adjacent SIP entities assuming delays of 1 ms per 100 miles. The INVITE service rate is fixed at 0.5 ms^{-1}. We observe that the mean response time is essentially linear with the arrival rate for the range of values considered. As expected, the mean response time increases with the mean propagation delay time. In our evaluated interval of arrival rates and propagation delays, the mean response time is in an acceptable range (as compared to the 100-300 ms for PSTN switches). Figure 5 shows the mean number of jobs in system as the arrival rate varies. We observe that the mean number of jobs is quite small (less than 10), even under propagation delays corresponding to a distance of 1000 miles.

We then compute these same measures of interest, this time varying the service rates for processing INVITE requests. Figures 6 and 7 show the mean response time and mean number of jobs in system as the service rate is varied. In this analysis, the arrival rate of INVITEs is fixed at 0.3 ms^{-1}; i.e. 1 million BHCA.





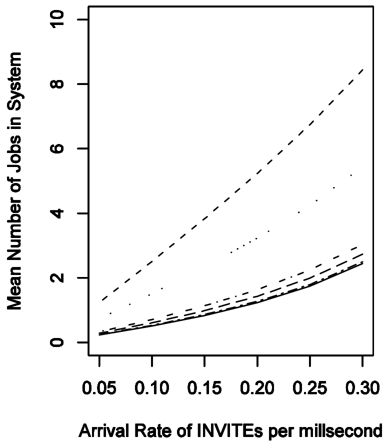
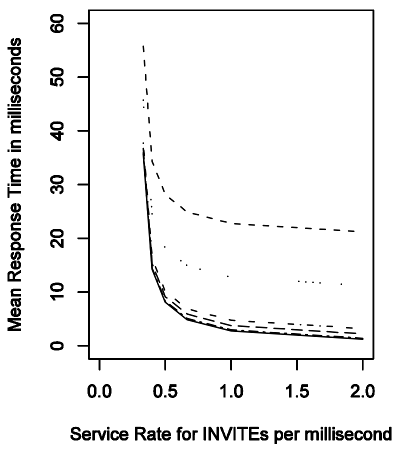


Fig. 4. Mean response time under varying arrival rates

Fig. 5. Mean number of jobs under varying arrival rates



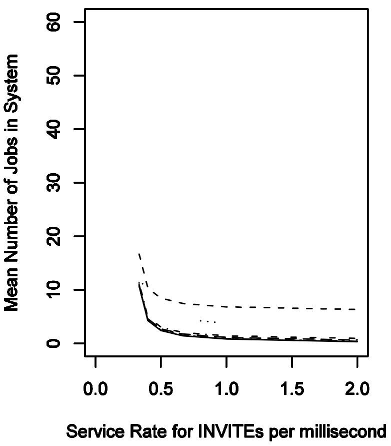


Fig. 6. Mean response time under varying service rates

Fig. 7. Mean number of jobs under varying service rates

4.3 Performance Model for Multiple SIP Servers

We now extend the model and analysis in two ways: first, to hosts running multiple proxy servers for scalability, and second, to a network of SIP proxy servers.

Multiple Proxy Servers on a Single Host

Clearly a single server solution for a proxy is not scalable. We therefore provide performance results for a multi-server proxy host. We extend the model of Figure 2 to queuing networks with the same structure, but with each M/M/1 queue replaced by a M/M/*m* queue. The equations for computing the mean response time and mean jobs in system are standard (c.f. [11]). Figure 8 depicts the performance results for the model of Figure 2 with M/M/*m* queues, where the number of servers, *m* is varied between 3 and 10, and the propagation delay is set to zero. The lower bound of 3 servers corresponds to the minimum number of servers needed to ensure that the queuing network is stable. A key observation from Figure 8 is that below a certain threshold for the service rate μ (i.e. 0.3 INVITEs ms^{-1}), the mean response time to process requests can grow significantly even under small changes in the service rate. Thus, this indicates the minimum service rate for multiple server hosts to ensure robustness of the proxy server. Our second observation is that for values of μ greater than this threshold, not only is the mean response time less sensitive to changes in the service rate, it is also largely independent of the number of servers in a single proxy server host. This implies that a small number of multiple servers with a service rate of 0.3 is sufficient, so large numbers of servers or faster service rates are not necessarily needed. Figure 9 depicts a similar analysis, where the mean network delay is fixed at 1 ms. The results are similar to Figure 8, with an increase in mean response time corresponding to the network delay.

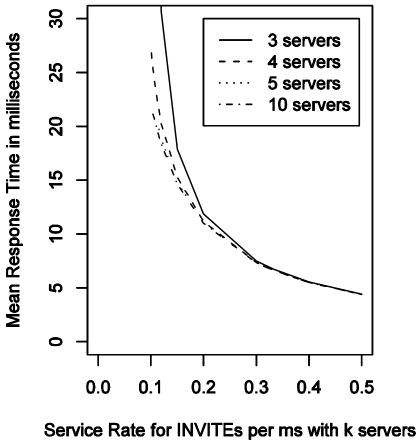


Fig. 8. Mean response time of multiple server host under varying service rates

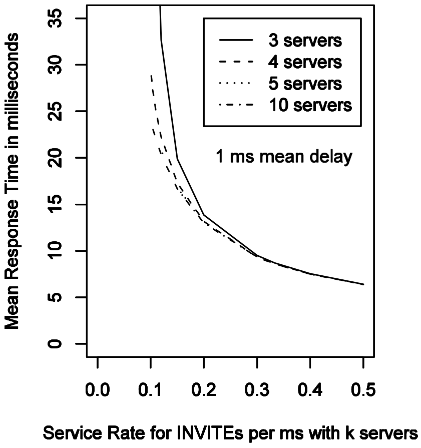


Fig. 9. Mean response time of multiple server host with varying service rates and network delay

Chain of SIP Proxy Servers

We next extend our analysis of a single server host in an orthogonal direction: namely, to a network of proxy servers modeling multiple hops in an end-to-end network. We thus extend our performance measures of interest of mean response time and mean jobs in system to reflect the end-to-end network. In particular, the mean end-to-end response time is defined as the mean elapsed time from the time t_1 an INVITE request from an User Agent Client (UAC) arrives at the proxy server until the time t_2 that the proxy server sends a final response to the UAC; this mean response time now includes the time taken by all the intermediate proxies and the far end UAS to set up the call. Similarly, the mean number of jobs in system is now defined as the mean number of calls being set up or waiting to be set up by any of the intermediate proxy servers involved in setting up the call.

In order to do this analysis, we need to recursively replace each station modeling the far end in our queuing network by a copy of the queuing network. However, separately replacing the fserv-180 and fserv-non-200 stations by copies of the queuing network is incorrect, since the arrival rate into the copies of the queuing network would recursively be a fraction (0.8 or 0.2) of the arrival rate into the base model. Hence, this recursive model would incorrectly assume greater capacity in the system. We thus first use an alternative model to our queuing network in which the fserv-180 and fserv-non-200 stations are replaced by a single fserv station. This model is depicted in Figure 10, where Nserv is the sum of the mean number of jobs at stations fserv-180 and fserv-non-200 computed from the model of Figure 2.

It is straightforward to show that, for any arrival rate λ, if the service rate μ_{fserv} is given as λ(Nserv+1)/Nserv, the mean response time and mean number of jobs of this alternative model and the original model are equivalent. We thus construct our model of SIP networks by recursive substitution into this alternate model. In particular, we recursively substitute the fserv station in this alternate model with a copy of this alternate queuing network, and then compute the mean end-to-end response time and mean number of jobs in the end-to-end system. A similar construction is done for the extended model that included propagation delays. Figures 11 and 12 show the results of this analysis, where the length of the proxy chain is varied from 1 to 6. The different lines again correspond to varying the propagation delays. The arrival rate is fixed at 0.3 ms^{-1} and the service rate for INVITEs is fixed at 0.5 ms^{-1}.

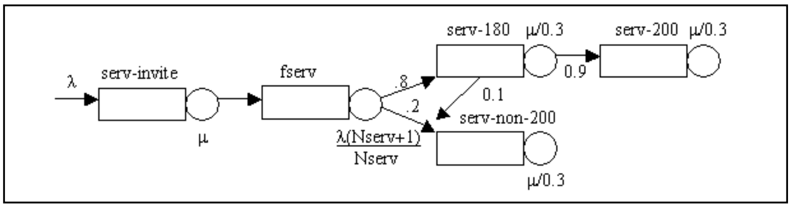
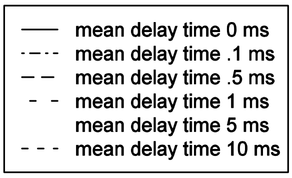


Fig. 10. Equivalent model for analysis



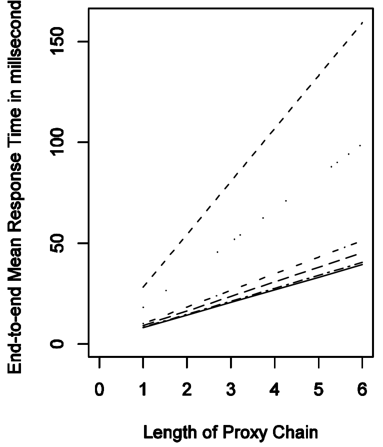


Fig. 11. End-to-end mean response time under varying length of proxy chains

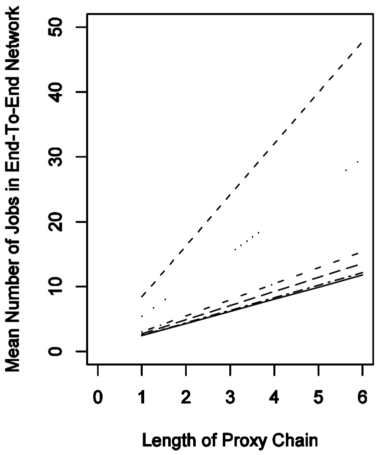


Fig. 12. End-to-end mean number of jobs under varying length of proxy chains

5 Reliability Analysis

The reliability metrics of interest for the proxy server are the steady-state system availability and the probability of job loss (i.e. loss of SIP call requests). We first develop a standard reliability model for the single proxy server for various replications schemes. The reliability model is then combined with the queueing performance model of Section IV to predict the probability of job loss for these replication schemes. For this analysis, we used the hierarchical reliability and performability models and associated closed form expressions for computing availability and loss probability presented in [5].

The existence of fault tolerance software running at the application layer, that provides process and node error detection, recovery and checkpointing capabilities (as appropriate), is assumed for the proxy server. As in [5], the server is assumed to exhibit fail-silent behavior. When there is a server failure, the messages at the server, could be lost or saved depending on the recovery mechanisms implemented. The same applies to new messages arriving at the server during the detection and recovery intervals. Since the queuing network performance model assumes an infinite size buffer for the wait queue, messages are not lost due to buffer overflow. Thus, in the event of a server failure, the following 3 message loss scenarios are of interest: queued jobs, in-service jobs and new job arrivals when the system is down are lost (Case V in [5]); queued jobs, in-service jobs and new job arrivals when the system is down are not lost (Case II in [5]); and queued jobs and in-service jobs are lost and new jobs arrivals when the system is down are not lost (Case VI in [5]).

5.1 Reliability Models

Continuous Time Markov Chain (CTMC) models, which capture the failure, error detection and recovery behavior of the server are evaluated for the following replication schemes: no replication, cold replication and warm replication. Server failures are caused by process or node failures, and it is assumed that there is only a single failure in the system at any time.

No Replication. There is a single proxy server with no fault tolerance software. Error detection and recovery are done manually. The unavailability of the server is observed only after the failure is detected and recovery is initiated after detection.

Cold Replication. There are two proxy servers running in active/cold standby mode with fault tolerance software at the application layer. Upon detection of a failure of a process in the active node, the process is restarted and the system is returned to a working state; with some probability this may require switchover to the standby node. Upon detection of a failure of the active node the recovery action is to switchover to the standby node. In this case the switchover time includes the time required to bring up the node. We follow the cold replication model given in [5].

Warm Replication. There are two proxy servers running in active/warm standby mode with fault tolerance software at the application layer. In the event of active process or processor failure, the standby node assumes the role of the active node after detection of the failure and switchover. A new backup is started on another available node. In the event of standby process failure, the process is restarted or, if it exceeds the threshold of restarts, it is started on a different node. We follow the warm replication model given in [5].

For all of the above replication schemes, availability is calculated from the pure reliability models by adding the steady state probabilities of the server up states. The pure reliability models at the high level and the queuing models of Section IV at the lower level are combined to compute the probability of call loss. In particular, rewards are associated with each state and transition of the reliability model. Rewards associated with a state reflect the rate of expected loss of call requests in that state; lost call requests accumulate at the specified reward rate during the expected time spent in the state. Impulse-based rewards associated with transitions reflect the number of calls lost when the transition takes place. Expected rate of loss is computed by the accumulation of lost call requests in states and during transitions; we use the closed form equations from [5]. As in [5], loss probability is calculated by dividing the expected rate of loss of incoming jobs by the expected job arrival rate.

5.2 Results of Reliability and Call Loss

The following parameter values (with exponential distributions) are assumed for the reliability and call loss analysis of SIP proxy servers:

Job arrival rate, $\lambda = 0.3$ ms^{-1}
Job service rate, u = 0.5 ms^{-1}
Process failure rate,$\gamma_p = 0.1$ day^{-1}
Node failure rate, $\gamma_n = 0.05$ day^{-1}

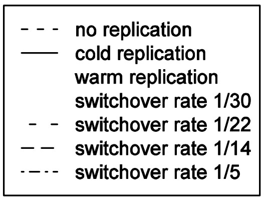
Process failure detection rate δ_p= 1 sec^{-1}
Manual recovery rate, $\tau = 1/120$ sec^{-1}
Process restart rate, $\tau_p = 1/30$ sec^{-1}
Process failover rate, $\tau_n = 1/120$ sec^{-1}

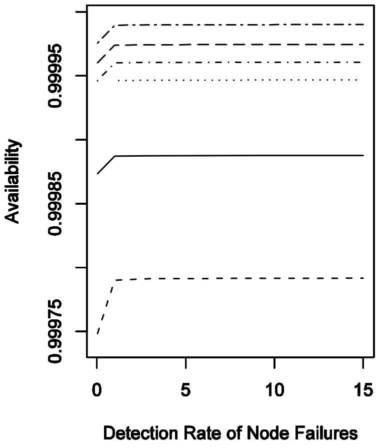
The node failure detection rate, δ_n is varied from 0.1 sec^{-1} to 15 sec^{-1}. The node switchover rate, $\tau_{s,}$ from failed to warm standby, is varied from 1/5 sec^{-1} to 1/30 sec^{-1}. The job (INVITE) arrival rate and service rate are as in the models of Section IV.

Figure 13 shows the availability of a proxy server for different values of the node failure detection rate for the case of no replication, cold replication and warm replication. Node availability greater than 0.9999 is achievable with warm replication and it is not sensitive to increases in the detection rate beyond 1 sec^{-1}.

Figure 14 shows the probability of job loss for a proxy server for different values of the node failure detection rate for no replication, cold replication and warm replication. The loss scenario is that queued jobs, in-service jobs and new job arrivals when the system is down are lost; therefore, no checkpointing is required. For all cases, there is an initial decrease in the probability of job loss as the node failure detection rate is increased from 0.1 sec^{-1} to 1 sec^{-1} and, for further increases in the detection rate, there is an increase in the probability of job loss except for the no replication case where it remains constant. The probability of message loss increases significantly for values of the detection rate greater than 12 sec^{-1} due to the increased overhead associated with the higher detection rate.

We assume that the buffer for job arrivals entering the system is of infinite size and, therefore, no jobs are lost due to buffer overflow. In Figure 15, we plot the expected number of job arrivals when the system is in the down state against the detection rate of node failures for different replication schemes. As expected, this figure is highest for the no replication case (longest downtime) and lowest for the warm replication case (shortest downtime). The results, however, are not sensitive to changes in the node failure detection rate beyond 1 sec^{-1}. Next, in Figure 16, we show the mean time required to service the jobs accumulated in the arrival queue (while the system was in a down state) as a function of the node failure detection rate. The service time for these jobs ranges from 75 seconds for no replication, 40 seconds for cold replication and 2 to 20 seconds for warm replication depending on the node switchover rate. The point to note is that, when the job arrival rate is high, saving job arrivals during the recovery interval is not worthwhile in call processing applications — the call setup delays for no replication and cold replication schemes would be unacceptable. This implies that checkpointing will not provide any benefits for the no replication and cold replication schemes.





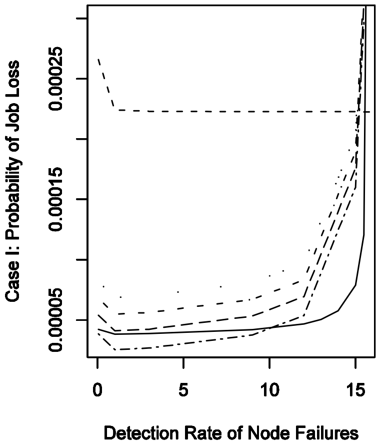
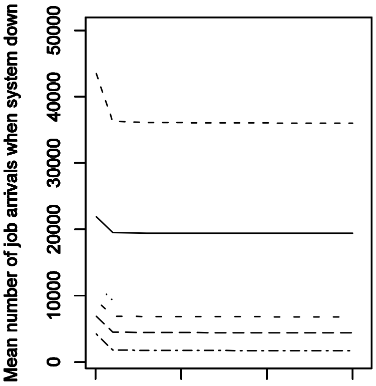


Fig. 13. Availability of proxy server

Fig. 14. Probability of message loss of proxy server



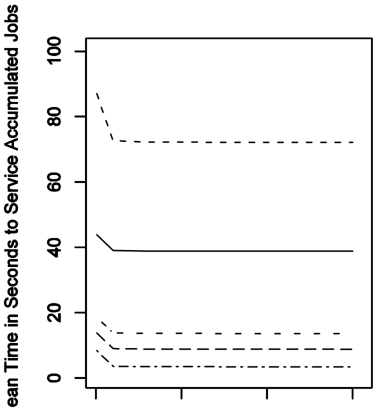


Fig. 15. Arrival of INVITE requests during down state

Fig. 16. Mean time to service jobs in arrival queue

6 Conclusions and Future Work

We have presented performance and reliability models for SIP networks and analyzed the behavior of the network under varying arrival rates, service rates, network delays, and replication schemes and associated failover rates. Key metrics that were analyzed include (end-to-end) mean response times, (end-to-end) mean number of jobs in the system, availability, probability of job loss, and mean time to process jobs that arrive when the system is down. Our analysis indicates three key findings. First, for an arrival rate of 1 million BHCA our results show the mean response time falls within an acceptable range, and that beyond a certain point, increases in service rates or number of servers on a single host do not yield significant improvements in mean response time. In particular, our results show that for single server hosts and service rates of 0.5 INVITE ms^{-1}, mean response times are less than 10ms. Furthermore, service rates greater than 1 ms^{-1} do not yield significant improvements in mean response time. Similarly, for multiple server hosts and service rates of 0.3 ms^{-1}, response times remain acceptable. Second, our results indicate that in steady state there are very few jobs in the system that are in a setup state. For example, in the steady state we observe that single server hosts with service rates greater than 0.5 requests per ms, there are no more than 10 jobs in the setup state in a single proxy server. For chains of single server proxies up to length 6, there are no more than 50 jobs in the setup state across all proxies in a SIP network. Given these results, we question whether it is necessary to add checkpointing in a SIP network. As noted earlier, however, if the delay representing the time taken by the user to answer the call is included in the analysis there will be more jobs in the system in a *ringing state*. Our future work will extend the performance analysis to multiple servers on hosts.

Third, our results demonstrate that saving incoming jobs when the system is down yields acceptable mean response times only under certain replication schemes. For no replication and cold replication, the mean time to service the INVITE requests accumulated during the recovery interval will require 40-75 seconds. Given that the normal lifetime of a SIP transaction is 32 seconds [1], saving job arrivals during the recovery interval is counter-intuitive. For warm replication, however, the mean time to service the jobs accumulated during the recovery interval is 2-20 seconds, depending on the value of the node switchover rate. For this replication strategy, one can consider saving new jobs that arrive during the recovery interval. However, since, as discussed above, calls in the setup state likely need not be saved, a comprehensive checkpointing strategy is not necessary. Our future work will also extend this aspect of our analysis with multiple servers on hosts.

The reliability model presented results for an assumed set of input parameter values. The results indicate that, to achieve the level of reliability in SIP networks that is comparable to PSTN, warm replication is required. In practice, these models can be used to determine required design targets such as switchover time and error detection time to achieve a given level of proxy server availability.

Future work will focus on validating the performance and reliability model parameters and results with lab measurements and field data. Additional future work includes relaxing assumptions about exponential distributions, including protocol timers in the model and extending the reliability model to multiple servers.

References

[1] J. Rosenberg, et al., "The Session Initiation Protocol (SIP)", *IETF RFC 3261*, June 2002, <http://www.ietf.org/rfc/rfc3261.txt>.

[2] J-S. Wu and P-Y Wang, "The performance analysis of SIP-T signaling system in carrier class VoIP network", Proceedings of the 17th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2003.

[3] Vemuri and J. Peterson, "Session Initiation Protocol for Telephones (SIP-T): Context and Architectures", IETF RFC 3372, September 2002, <http://www.ietf.org/rfc/rfc3372.txt>

[4] H. Schulzrinne, et al., "SIPStone - Benchmarking SIP server performance", April 2002, <http://www.sipstone.org/files/sipstone_0402.pdf>.

[5] S. Garg, et al., "Performance and Reliability Evaluation of Passive Replication Schemes in Application Level Fault Tolerance," Proceedings of the 29th Annual International Symposium on Fault-Tolerant Computing, Madison, WI, June 1999.

[6] J. F. Meyer, "On evaluating the performability of degradable computing systems", IEEE Transaction on Computers, Volume 29, No. 8, pp. 720-731, August 1980.

[7] T. Russell, "Signaling System #7", (Second Edition), McGraw-Hill Publishing Company, 1995.

[8] G. Camarillo, "SIP Demysitified", McGraw-Hill Publishing Company, 2001.

[9] J. Davidson, et al., "Voice over IP fundamentals", Cisco Press, 2000.

[10] Zhu, "Analysis of SIP in UMTS IP Multimedia Subsystem", MSc. Thesis, Computer Engineering, North Carolina State University, 2003.

[11] R. Jain, "The Art of Computer Systems Performance Analysis", John Wiley and Sons, Inc., 1991.

[12] F. Lipson, "Verification of Service Level Agreements with Markov Reward Models," South African Telecommunications Networks and Applications Conference, September 2003.

[13] P. Barford and J. Sommers, "Comparing Probe- and Router-Based Packet-Loss Measurements," IEEE Internet Computing, Vol. 8, No. 5, pp. 50-56, September-October 2004.