International Journal of Engineering Research
Volume No.5 Issue: Special 5, pp: 992-1128

ISSN: 2319-6890) (online), 2347-5013(print)
20 May 2016

SIP signaling system with timed-out packets Modeling and simulation based on SimEvents

Garima Mishra
Department of Electronics and Communications,
HKBK College of Engineering,
Bangalore 560045
garima.misra80@gmail.com

Abstract- Session Initiation Protocol (SIP) has retransmission mechanism which in addition to reliability also increases the load in the network by creating redundant messages. SIP has a built-in mechanism to control overload which is not sufficient to handle extreme load situations. Hence, we propose to model a hop-by-hop overload control algorithm to take care of this issue. This paper describes the modeling of SIP proxy servers which controls the number of redundant messages in the system. The simulation results let us analyze the system performance under different load conditions and the results of the proposed model are compared with the standard SIP as per RFC 3261.

Keywords- SIP signaling,Overload control, Retransmission rate, Loss-based overload control algorithm

I. Introduction

The Session Initiation Protocol (SIP) is proposed as a signaling protocol for all Internet Protocol (IP)-based networks by Internet Engineering Task Force (IETF) network working group. The 3rdGeneration Partnership Project (3GPP) has adopted SIP as the basis for establishing and modifying the IP connectivity for its IP Multimedia Subsystem (IMS) architecture. SIP is designed such that it is independent of the underlined transport layer. So, its messages can be sent over Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) [1]. As we know that UDP provides the simplest transport but provides no congestion control. SIP has reliability mechanism and allows the use of UDP. For SIP over UDP there is a possibility of message loss hence, a series of lost packets on a heavily loaded system can cause retransmissions, which in turn produces more packets into the system and finally introduces more delay to the system.

In IP-telephony converged networks, all requests exchanged are time limited. Therefore, SIP signaling must process the session setup requests within the real-time constraints. The main processing requirements of SIP elements are examined by [2]. In their study those factors are described which determine the cost congestion and delay to the network. The various reasons that cause server loading and degrades the system performance is also discussed in RFC 5390 [3].

A study on congestion control by performing simulation techniques is done by many authors [4],[5],[6] in past and they have also proposed overload control algorithms to increase the performance of SIP network for real time applications. The comprehensive study of the current SIPand its performanceunder different load conditions is given by [5]. The author proposed a distributed overload control mechanism and evaluated it using simulations.

The complete mathematical modeling and its analysis based on the simulation model presented in this paper has already been presented in [7]. The objective of this paper is to discuss about the retransmission mechanism in SIP and modeling SIP network elements such as, SIP message generators, SIP proxy and the channel using SimEvents, a discrete event simulation toolbox. An overload control algorithm is implemented inside the SIP proxy to mitigate the redundant messages into the system. The results of the proposed model are also compared with the standard SIP system.

The rest of this paper is organised as follows. In sectionII, SIP call setup procedure and retransmission mechanismare discussed. In section III, modified queueing system withtime-out packets is developed and an overload control algorithmis proposed. The simulation using SimEvents is explained in section IV and simulation results are given in section V. The paper is finally concluded in section VI.

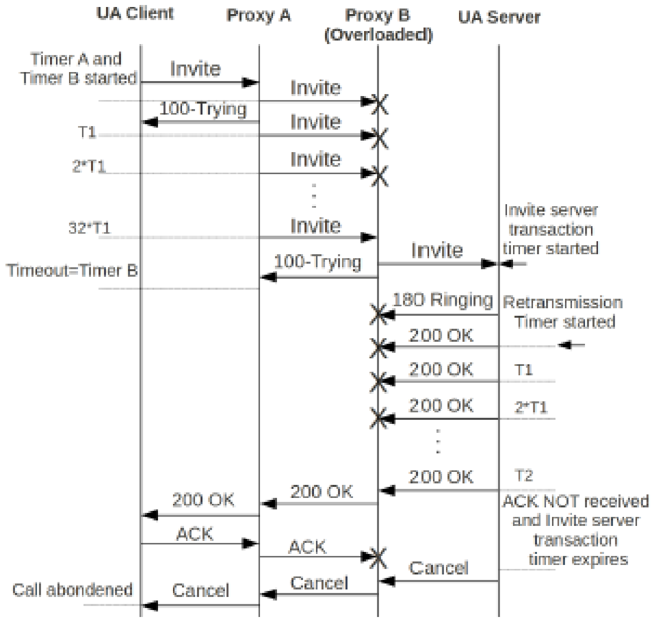

Fig. 1: SIP message retransmissions

II. SIP Session Establishment Procedure and its Retransmission Mechanism

*A. SIP session establishment:*SIP is used to establish, update and terminate multimediasessions. A session establishment using SIP requires three-wayhandshake of signaling messages. An example SIP messageflow involving two user agents (UAs) and two stateful proxies(Proxy A and Proxy B) is shown in the Fig. 1 with fewretransmissions. As per the shown SIP call setup scenario,User Agent Client (UAC) sends an *Invite* request to theintermediate proxies, which forwards this request to the UserAgent Server (UAS). The final acknowledgement response *ACK* is sent after receiving few provisional responses fromUAS for an *Invite* message. The call setup procedure isconsidered successful once final *ACK* is sent from UAC toUAS. Hence, the total delay in session setup is considered bythe instant first *Invite* is sent and SIP transaction timer startstill the *ACK* is received or timer B expires.

*B. SIP Retransmission Mechanism :*The Fig. 1 shows a scenario of SIP message retransmissionsin case one of the SIP proxies is overloaded. When a newrequest (*Invite*) is sent by a UAC or a stateful proxy, thena request retransmission timer (Timer-A) and a transaction(Timer-B) are started. These SIP timers scale with *T*1 where*T*1 is supposed to be an estimate of the round trip time (RTT)of the request and its default value is 500*ms*. If no response tothe request (as identified via response containing the identicallocal tags, remote tags and specific SIP headers such as call-IDand Cseq) is received when *T*1 expires, the request is re-sent.

The first informational response (1*xx*) is received by the UAC confirms receipt of the *Invite* and stops retransmission. For this reason, a hop-by-hop 100*Trying* response is used to just minimize the retransmission of the *Invite* in the network. Astateful proxy stores forwarded request for maximum 32*secs*. Hence, the transaction timer expires after 32 seconds. For example, if *Timer-A* is 500*ms* at the start then there are 6 retransmissions making total 7 redundant messages in the system including the first request. The retransmissions occur at an interval of 500*ms*, 1*s*, 2*s*, 4*s*, 8*s*, 16*s*,32*s*. For*Non-Invite* transactions retransmissions continue at an interval of *T*2 even though a provisional response is received. The default value of *T*2 is 4*sec*. Now, the retransmission occurs at every *T*2 interval. This capped exponential process continues until a maximum of 10 retransmissions are reached. Refer Table-4 in RFC 3261 [8] for a summary of SIP timers. The hop-by-hop *Invite*-100*Trying* transaction is the major workload contributor in case of overload. Therefore, we focus on the hop-by-hop *Invite*-100*Trying* transaction in this paper. We represent the SIP proxy servers as the queueing system with impatient customers. First, equivalence between a class-*k*multi-server queue and *M/G/*1 queue with multiple vacationsis shown and then secondly, the effect of introducing impatient customers is studied using simulations.

III. Queueing System with Impatient Customers

The effect of retransmitted request and their timed-out responses is captured in the queueing system with impatient customers. SIP proxy servers may waste their processing time and resources for serving an already timed-out message that will finally be rejected. Therefore, we propose to easily discard these timed-out messages. Each arriving message wait for their service for a limited time only and should leave the system if not served during the specified fixed time τ. The discarded messages may either leave the system and become a lost message or may go for a retrial. So, now we consider the traditional *M/G/*1 queueing system as a *M/G/*1 queueing system with vacations but with fixed time-out. In this study, only new requests and their responses are considered for time-out because assuming other messages (*Non-Invite*) as impatient customers may lead to premature termination of already on-going sessions. We apply concept of impatientcustomers to the timed-out SIP messages present in the system. When the estimated RTT of response messages exceed their fixed time limit τ= [500*ms;* 1*s;* 2*s;* 4*s;* 8*s;* 16*s;* 32*s*], these timed-out messages are dropped or discarded from the system even before further processing. These timed-out messages are considered as impatient customers in the queueing system.

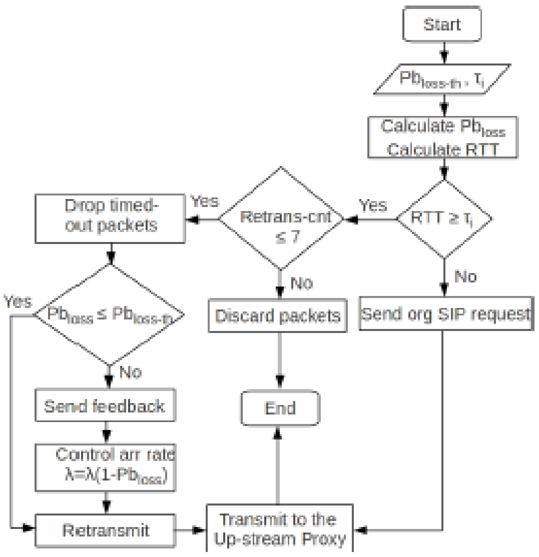

Fig. 2: Flow chart for overload control algorithm

A. Hop-by-hop Overload Control: In this section, an explicit

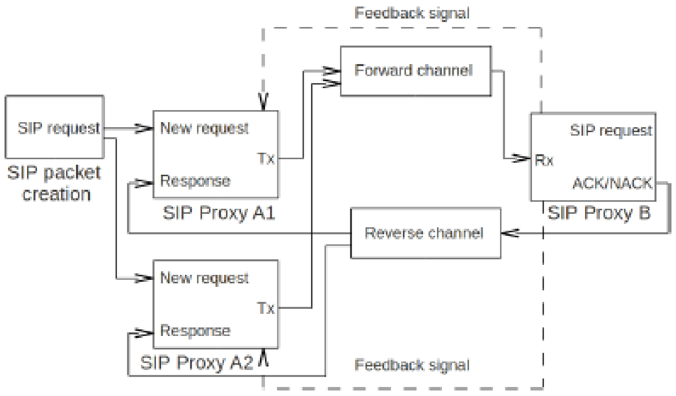

Fig. 3: End-to-end SIP signaling simulation model

overload control algorithm is developed by applying timeout mechanism over the queueing system. The above mentioned time-out approach is applied in SIP proxy servers to reduce overload by controlling redundant messages present in the system. As shown in the Fig. 3, a downstream SIP proxy, (proxy B) receives signaling traffic from multiple upstream sources (proxy A1,proxy A2). When downstream proxy gets overloaded,it sends a feedback signal to its corresponding downstream proxy to adjust the sending rate of the traffic by a certain amount. We propose an algorithm to calculate this percentage based on the calculation of packet loss rate in the system using previously discussed time out approach. The main task of the modified system is to reduce the redundant messages in the system. The control unit in the downstream proxy keeps on calculating Pb_{loss}at certain time instants and once the Pb_{loss}crosses its threshold value $Pb_{loss}th$ then the feedback signal is sent. The proposedoverload control (OC) algorithm is summarized in the flow chart shown in the Fig. 2.

IV. Model Implementation in SimEvents

The proposed queueing model with packet time-out is modeledand simulated using SimEvents.SimEvents is a discrete event simulation (DES) tool to design a real SIP signaling network [ix].

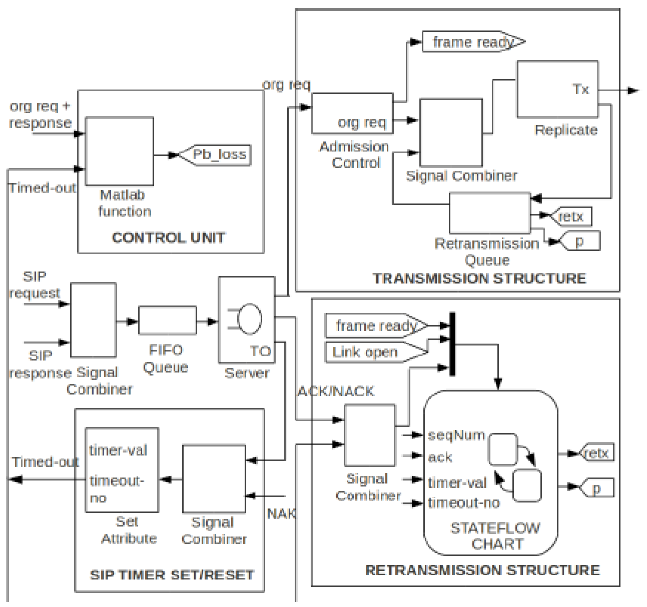

Fig. 4: The main components of SIP proxy modeled in Simevents

1

The information source represented by the packet generation subsystem generates SIP signaling requests which represent SIP Invite requests from multiple SIP UAs. The multiple requests are generated using a time-based generator block at different rates according to a constant, uniform, exponential and user-defined inter-generation time and each request has a set of attributes like, unique sequence number that identifies a session, initial SIP timer value and a tag that makes the original SIP requests different from the retransmitted requests. The SIP proxy has 2 functionalities; it first receives SIP requests and advances them to the next SIP proxy server through the forward communication channel. Also, it does not issue new SIP requests but retransmits those messages which do not get acknowledged with an ACK response. The second functionality is that the SIP proxy also responds forthe received requests with an ACK to the correspondingdownstream proxy. The forward/reverse channel representthe delay on the link. It is assumed that the packets maybe corrupted in the channel but not dropped. The reversechannel is assumed to be error-free. The main componentsof SIP proxy are shown in Fig. 4. There are 4 blocks, thetransmission block, which is responsible for forwarding SIPmessages to the next upstream proxy. The second block is the retransmission block, which takes care of the retransmission of the SIP messages based on their response (e.g. NACK) or timed-out packets in the system.

The SIP proxy uses a state-flow chart to control the retransmissions of SIP messages. It can have one of these states and the next state is defined by the state parameters: *Transmitting state*: placing new request on the link (p=1,retx=0,timer-val=1,timer-no=1),*Retransmittingstate*:Retransmitting SIP messages with timeout or NACK,*Wait state*: wait till the link is available, *Idle state*: No newdata is present to be transmitted. The third block is the SIPtimer block which sets or resets the SIP timer values as pergiven in the RFC-3261. And the fourth block is the controlblock which is developed to control the aggregate arrival rate to the proxy based on the calculated signaling packet loss rate (Pb_{loss}). This calculated $Pb_{loss}$is sent as a feedback to the downstream proxy or a UA to control their sending rate. The above mentioned blocks model the SIP proxy which adheres to the functionalities mentioned in RFC 3261. The proposed algorithm mentioned in the previous section is modeled as a control unit of the transmitter block which calculates the probability of loss depending on the current total arrival rate and the timed-out messages in the system. Hence, incoming arrival rate from the downstream servers are regulated based on the calculated Pb_{loss}.

V Performance Evaluation and Simulation Results

In this section, discrete event simulations using SimEvents are conducted to verify our proposed algorithm for reducing redundant messages in the SIP signaling network. The simulations are done to observe the behavior of the overloaded SIP proxies and its after effects on the down-stream proxies. The following default parameter values are used, unless otherwise indicated; the arrival follows Poisson process with the arrival rate 8 and the mean service time 1/μ= 0.1 sec. The offered traffic utilization of 0.8 Er/ch is considered high enough to bring system under traffic overload state. Now, we shall analyse the advantage of including timeout control for new session setup requests. We consider the SIP proxy as *M/G/*1 queue with fixed timeout values. These fixed timeout values (τ= [0:5*s*; 1*s*; 2*s*; 4*s*; 8*s*; 16*s*; 32*s*]) are from standard SIP timers specified in [8]. In our simulations SIP is modeled as per the given standards by SIP RFC 3261 [8]. In the subsequent simulation results we have shown the comparison between the standard SIP proxy and the proxy with OCalgorithm.

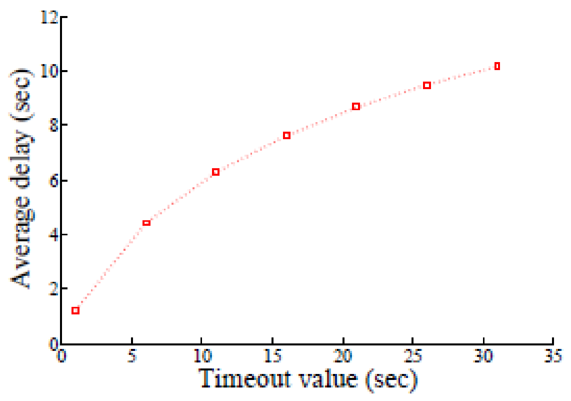

Fig. 5: Effect of time-limit values on average delay

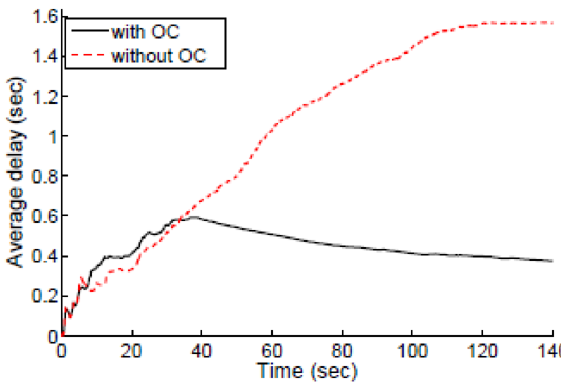

Fig. 6: Average end-to-end delay in case of overload in the system with and without OC algorithm

A. End-to-end delay

As described in the previous sections, the most significant performance measure for SIP session establishment is an end-to-end delay given that all the intermediate SIP proxies are loaded heavily. The average delay in queue of a served or lost messages, with and without timeout is shown in Fig. 6. It is observed in Fig. 5 that the average delay in the queue with timeout always remains smaller than the system with timeout. Also it is seen that the waiting time with timeout always remains much less than the specified timeout values 'τ'. As mentioned that in SIP there is a limit to the delay before transmitting the request. Hence, modified model with timeout behaves much better for the delay-limited applications.

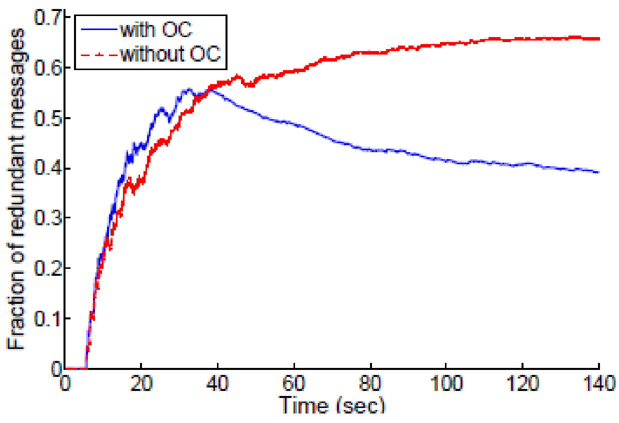

Fig. 7: Fraction of redundant messages in the SIP network with and without OC algorithm

B. Fraction of redundant messages

We intend to reduce the number of redundant messages in the subsystem which is caused by long queueing delays. Therefore, we first calculate RTT of the SIP messages, keepingtransmission and propagation delays negligible. Then using these RTT values in OC algorithm, dropping of timed out packets and retransmission decisions are taken. The Fig. 7 shows the fraction of redundant messages in the overloaded SIP server. We observe from the simulations that the proposed algorithm regulated the retransmission rate. There is a significant 30% decrease in the fraction of redundant messages in the system. The $Pb_{loss}$is calculated at the overloaded proxybased on the lost timed-out packets. The OC algorithm isactivated when the Pb_{loss}reaches its target value. The OC algorithm regulates the transmission rate of the corresponding down-stream proxy depending on the current Pb_{loss}value.

VIConclusion

The main goal of this paper is to demonstrate the effect of applying time-out approach to reduce SIP overload using SimEvents simulation toolbox. The performance results obtained from the simulations and their analysis provide some insights on system load condition factors where overload control becomes most efficient. The results show that the proposed overload control algorithm becomes useful for the congested IP links and loaded proxy servers. It is observed that the delay in the network is improved when SIP messages are queued with a fix time-limit. Also there is a reduction in the percentage of redundant messages in the system.From the performance results , we observe that SimEvents make a significant contribution to develop a complex system of discrete/continuous processing.

References

[i] G. Camarillo and M. A. Garca-Martn, *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds*, 3rd ed. Wiley & Sons, 2008.
[ii] M. Cortes, J. R. Ensor, and J. O. Esteban, "On sip performance," *Bell Labs Technical Journal*, vol. 9, no. 3, pp. 155–172, 2004.
[iii] J. Rosenberg, *Requirements for Management of Overload in the Session Initiation Protocol*, ser. RFC 5390 (Informational).Internet Engineering Task Force, Dec. 2008. [Online]. Available:http://www.ietf.org/rfc/rfc5390.txt
[iv] E. Noel and C. R. Johnson, "Novel overload controls for SIP networks." France: 21st International Teletraffic Congress, 2009, pp. 1–8.
[v] V. Hilt and I. Widjaja, "Controlling overload in networks of SIP servers," in *2008 IEEE International Conference on Network Protocols*. IEEE, Oct. 2008, pp. 83–93.
[vi] E. M. Nahum, J. Tracey, and C. P. Wright, "Evaluating sip server performance," *SIGMETRICS Perform. Eval. Rev.*, vol. 35, no. 1, pp. 349– 350, Jun. 2007.
[vii] G. Mishra, S. Dharmaraja, and S. Kar, "Reducing session establishment delay using timed out packets in sip signaling network," *InternationalJournal of Communication Systems*, 2014. [Online]. Available: http://dx.doi.org/10.1002/dac.2824
[viii] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, *RFC 3261 - SIP: Session Initiation Protocol*. IETF, 2002.
[ix] M. A. Gray, "Discrete event simulation: A review of simevents," *Computing in Science & Engineering*, vol. 9, no. 6, pp. 62–66, 2007.
[x] G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi , *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, 2nd ed. Wiley, 2006.