Effectiveness of SIP Messages on SIP Server

Abhishek Bansal[#1], Prashant Kulkarni[*2], Alwyn R. Pais[#3]
[#]Department of Computer Science & Engineering, NITK, Surathkal, India
[*]Supercomputer Education and Research Center, IISc, Bangalore, India
[1]abhishek.bansal.foru@gmail.com, [2]prashant.r.kulk@gmail.com, [3]alwyn.pais@gmail.com

***Abstract* — Voice over Internet Protocol (Voice over IP, VoIP) is one of a family of communication protocols, and transmission technologies. It is used for delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks. Session Initiation Protocol (SIP) is a signaling protocol, widely used for controlling multimedia communication sessions such as voice and video calls over Internet Protocol (IP). There are several DoS attacks by which we can disturb SIP server. In this paper, more importance has been given to DoS attack by flooding of different SIP-messages. A little work is done to analyze the performance of SIP server and quality of ongoing VoIP calls under DoS attacks. We show the utilization of CPU and memory during the multiple simultaneous calls. We have done our study using a customized analysis tool that has the ability to synthesize and launch flooding of different SIP messages. We define the performance metrics to measure the quality of VoIP calls under DoS attack. We have developed some programs and integrated them in a real SIP test bed environment to measure the performance of SIP server and quality of VoIP calls under DoS attack. Our measurements show that a standard SIP server can be easily overloaded by simple call requests. It also shows that simple call request can degrade quality of ongoing calls.**

***Index Terms* — VoIP, SIP, DoS.**

I. Introduction

H.323 [1] and SIP [2] are two major protocols, used to provide VoIP services. H.323 is the standard of International Telecommunication Union (ITU). SIP is proposed by Internet Engineer Task Force (IETF) [4]. SIP is an application layer signaling protocol. It is used to set up, modify and teardown the media sessions between two or more participants [5].

Denial-of-Service (DoS) attacks are explicit attempts to disable a target thereby pre-venting legitimate users from making use of its services. DoS attacks continue to be the main threat facing network operators [8].

The impact of a DoS attack depends on the target. If a particular client is a target then it can lead to denying the service to this user. But when a SIP server is the target, then it brings down the server. In this case, no user can get service [6]. Due to this attack, the provider's reputation also suffers. As a result, the provider may lose some of his existing and potential customers [7].

In this paper focus is given to DoS attack by flooding of different SIP messages. The mitigation of DoS attack is not within the scope of this paper. In our study, we try to investigate a number of relevant issues:

- Impact of flooding DoS attack on SIP-server.
- Impact of flooding DoS attack on quality of VoIP calls.

The rest of the paper is organized as follows. Section II introduces the security threats. These are used to analyze the robustness of SIP server. It also introduces the different performance metrics that we have defined to analyze the quality of VoIP calls. A little discussion is done on our attack synthesis and analysis tool. In Section III, we discuss the real test-bed which we have deployed for our experiments. We discuss and analyze results of our robustness study in Section IV. In Section V, we conclude our paper with an outlook to future research.

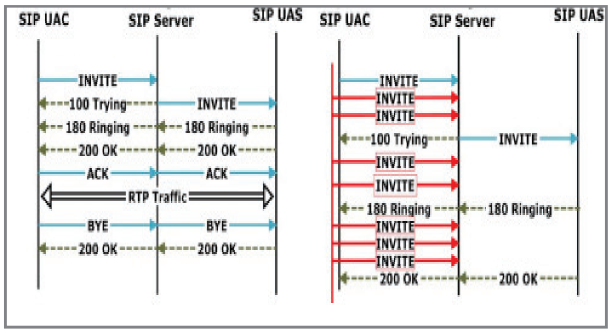

Fig.1. Call flow under DoS attack

II. Security Evaluation, Metrics And Attack Synthesis/Analysis Tool

In this section, we first describe the security threats. Then we define metrics, that are being used to measure the performance of SIP-server and quality of VoIP calls. Afterwards, we describe tools that are capable of launching flooding based DoS attacks and calculating the performance of the SIP servers in terms of these metrics.

A. *SIP Security Threats*

The easiest way to launch Denial of Service (DoS) attacks on a SIP proxy server is to flood it with a large number of unwanted calls requests [3]. As a result, its resources like internal memory buffers and CPU are exhausted and SIP-server cannot provide service even to the legitimate users (see Figure 1) [3]. In our experiments, 10 INVITE packets per second are sent to SIP server. The intensity of flood attacks varies from 100 INVITE packets to 1500 INVITE packets and different scenario are tested with INVITE-BYE, INVITE-

CANCEL, MESSAGE-RESPONSE, REGISTER-RESPONSE, NOTIFY-RESPONSE, OPTION-RESPONSE SIP messages.

B. *Performance Metrics*

Performance metrics are divided in two part, SIP-server metrics and VoIP call quality metrics.

SIP server metrics are used to check performance of server. These metrics are:

1) *CPU utilization*: CPU utilization is the average CPU usage of the machine, hosting a SIP server.

2) *Memory utilization*: Memory utilization is the average CPU usage of the machine, hosting a SIP server.

Metrics related to quality of VoIP help us to determine how much degradation in quality under DoS attack. These metrics are:

1) *Packet loss*: Number of packets at sender site - Number of packets at receiver site.

2) *Out of order packets*: These are packets which come later in call. These packets should be dropped because there is no use of these packets in real time transmission.

3) *Delay*: It shows time, taken by a packet to reach from sender to receiver. It is obtained by epoch time of packets.

$$\text{Delay} = \text{epoch time of receiver} - \text{epoch time of sender}$$

4) *Jitter*: Jitter [10] is defined as a variation in the delay of received packets.

$$J(i) = J(i-1) + (|D(i-1,i)| - J(i-1))/16$$

In the jitter estimator formula, the value D(i-1, i) is the difference of relative transit times for the two packets. The difference is computed as:

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i)$$

S_i is the timestamp from the packet i and R_i is the time of arrival for packet i [11].

C. *Attack Synthesis/Analysis Tool*

Now we describe the tools that we use to conduct our experiment.

1) *Asterisk*: It is a complete PBX in software. It is designed to interface any piece of telephony hardware or software with any telephony application [12].

2) *EKIGA Soft-phone*: EKIGA is a VoIP and video conferencing application. It supports many high-quality audio and video codecs.

3) *SIPp Tool*: This can be used to send multiple SIP messages together as well as single messages. The SIP Message should be written in XML format and the required parameters like extension details and authentication details in the csv file. The tool can be invoked using the command, `$SIPP -sf xmlFile serverIP:port -i clientIP -p clientport -inf csvfile -trace_err -m noOfTimes` [9].

4) *Attacker UAC and UAS*: It can be used to generate different message sequence and flooding of messages on server. It has two interfaces SIP-UAC and SIP-UAS.

5) *Wire-shark*: Wire-shark is a free and open-source packet analyzer. It is used to capture the packets.

6) *Top command*: Top command in Linux console shows CPU and memory utilization taken by different processes.

III. Experimental Test Bed

Now, we describe our experimental test-bed that we have used to evaluate the performance of SIP server and quality of VoIP calls under different types of attack scenarios. We have set up different test bed of different scenario. Every test bed contains the following major part:

A. *SIP UACs and UASs*

The benign users of the system are simulated as SIP UACs (callers) and UASs (cal lees). These clients are implemented using a modified version of SIPp, Attacker UAC and UAS, EKIGA. Actual calls are initiated by EKIGA. Flooding scenario is implemented by SIPp tool and attacker UAC. Simultaneous calls are simulated by SIPp tool.

B. *SIP Server*

We have selected asterisk as SIP-server. Asterisk is open source PBX machine.

C. *Analysis Machine*

To analyze CPU and memory performance, we have used top command on Linux console. We have used wire-shark to capture the RTP packets on sender and receiver machine. We have implemented some program to measure the quality of VoIP calls. We have used jNetPcap [13] library to capture, analyze RTP packets and hence deduce voice quality metric.

D. *Hardware Description*

The test bed has been designed in SERC lab at IISc. The experiments are done on an SERC network. The machines hosting SIP-server is Intel® core 2 duo 1.67 GHz processors with 2 GB RAM and 160 GB disk drives. The machines hosting UACs, UASs nodes are Intel(R) Pentium® 3.20 GHz processor with 1 GB RAM and 40 GB disk drives. All machines run Ubuntu OS.

The following figures are having different test bed scenarios for different testing.

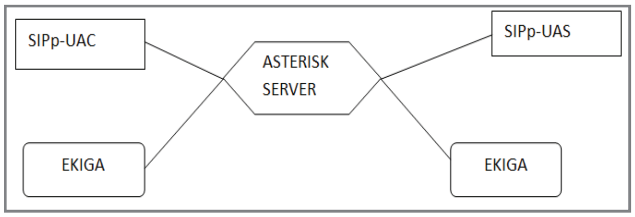

Fig.2. Stress testing asterisk

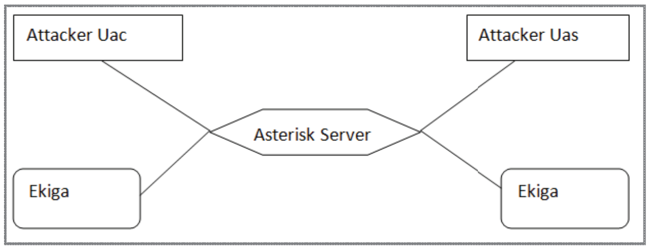

Fig.3. Attacker UAC-EKIGA test bed to check the performance of CPU, memory and quality of VoIP calls under different DoS attack.

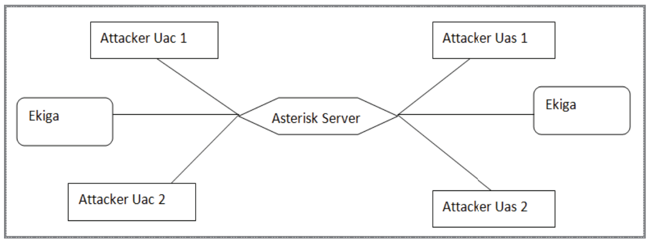

Fig.4. Attacker UAC-EKIGA test bed.

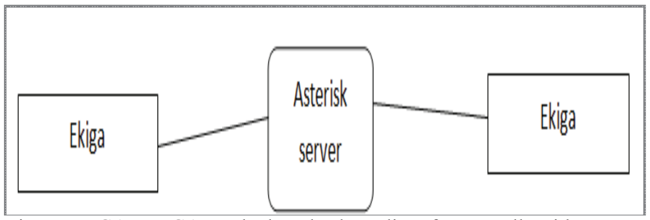

Fig.5. EKIGA-EKIGA test bed to check quality of VoIP calls without any attack.

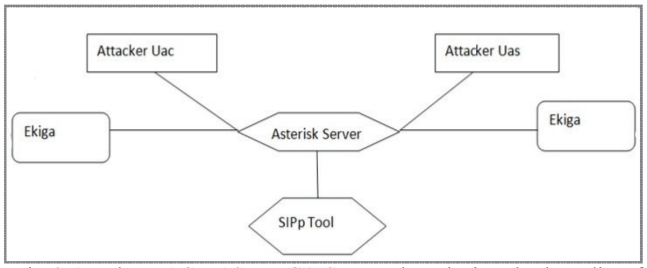

Fig.6. Attacker UAC-UAS-EKIGA-SIPP tool test bed to check quality of VoIP calls under different DoS attack.

IV. CPU and Memory Performance and Quality of VoIP Calls Result

We now present and analyze the performance of CPU, memory and quality of DoS attack during different flooding DoS attack by many graphs.

A. CPU and Memory Performance

During the stress testing of asterisk server, we increase the number of simultaneous calls and check the CPU and memory performance according to figure 2. From figure 7, it is clear that memory and CPU utilization increases as number of calls increases. But a maximum number of calls depend upon hardware configuration and number of active files descriptors. In our experiment, maximum number of calls is 1387 after which initiating call fails. Asterisk server gets crashed after bombarding it with more messages.

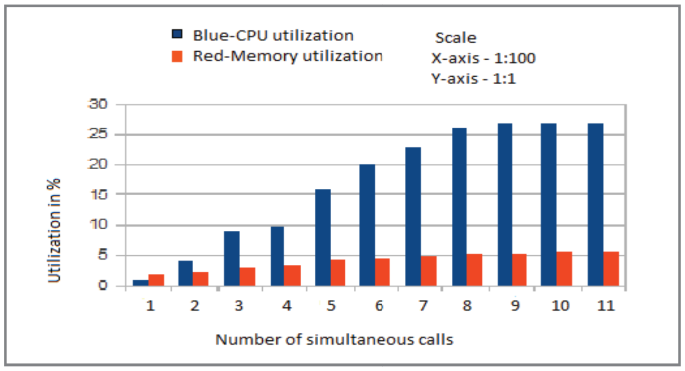

Fig.7. Stress testing asterisk

Now we use the attacker UAC tool to generate different message sequence. According to figure 3, first we use one attacker UAC and UAS to generate the flooding of different SIP messages at different rates and check the performance of CPU and memory. Here we find that impact of flooding by this tool on CPU and memory is negligible. Table 1 represents analysis of CPU and memory utilization for various message sequence at different rates, time elapsed (MINUTE: SECOND) and also represent that, can a new call be initiated during DoS attack.

Message Scenario	Number of packets	Time elapsed (M:S)	CPU	Memory	New Call
INVITE-BYE	1000	0:50	0	.02	YES
INVITE-BYE	2000	1:40	0	.02	YES
INVITE-CANCEL	1000	0:34	0	.02	YES
INVITE-CANCEL	2000	1:06	0	.02	YES
REGISTER	1000	0:22	0	.02	YES
REGISTER	2000	0:43	0	.02	YES
OPTION	1000	0:20	0	.02	YES
OPTION	2000	0:43	0	.02	YES
MESSAGE	1000	0:21	0	.02	YES
MESSAGE	2000	0:42	0	.02	YES
NOTIFY	1000	0:22	0	.02	YES
NOTIFY	2000	0:44	0	.02	YES

Table1. CPU and memory performance during DoS attack

Now we use two different attackers UAC and UAS to do more flooding as shown in figure 4. Each attacker UAS bombards1000, 2000, 5000, 10000 packets of different message sequence. We analyze the CPU and memory performance during bombarding. We also verify whether new call could be made when test bed is under attack and CPU, memory utilizations are also measured. Here time1 and time2 represent elapsed time in (MINUTE: SECOND) during flood by one attacker and two attackers respectively. Table 2-7 show results.

INVITE-BYE							
No. of packets	CPU Utilization	Memory Utilization	Time1 (M:S)	Time2 (M:S)	EKIGA Working		
					CPU	Memory	Call
1000	0	1.8	0:50	0:50	1.0	1.8	YES
2000	0	1.8	1:40	1:41	1.0	1.8	YES
5000	0	1.8	4:0	3:54	1.0	1.8	YES
10,000	0	1.8	7:55	8:0	1.0	1.8	YES

Table2. INVITE-BYE call scenario

INVITE-CANCEL							
No. of packets	CPU Utilization	Memory Utilization	Time1 (M:S)	Time2 (M:S)	EKIGA Working		
					CPU	Memory	Call
1000	0	1.8	0:32	0:33	1.0	1.8	YES
2000	0	1.8	1:04	1:04	1.0	1.8	YES
5000	0	1.8	2:41	2:42	1.0	1.8	YES
10,000	0	1.8	5:19	5:18	1.0	1.8	YES

Table3. INVITE-CANCEL call scenario

REGISTER-REQUEST							
No. of packets	CPU Utilization	Memory Utilization	Time1 (M:S)	Time2 (M:S)	EKIGA Working		
					CPU	Memory	Call
1000	0	1.8	0:20	0:20	1.0	1.8	YES
2000	0	1.8	0:43	0:43	1.0	1.8	YES
5000	0	1.8	1:47	1:47	1.0	1.8	YES
10,000	0	1.8	3:39	3:40	1.0	1.8	YES

Table4. REGISTER-REQUEST call scenario

MESSAGE-RESPONSE							
No. of packets	CPU Utilization	Memory Utilization	Time1 (M:S)	Time2 (M:S)	EKIGA Working		
					CPU	Memory	Call
1000	0	1.8	0:22	0:22	1.0	1.8	YES
2000	0	1.8	0:42	0:43	1.0	1.8	YES
5000	0	1.8	1:48	1:49	1.0	1.8	YES
10,000	0	1.8	3:38	3:37	1.0	1.8	YES

Table5. MESSAGE-RESPONSE call scenario

NOTIFY-RESPONSE							
No. of packets	CPU Utilization	Memory Utilization	Time1 (M:S)	Time2 (M:S)	EKIGA Working		
					CPU	Memory	Call
1000	0	1.8	0:26	0:27	1.0	1.8	YES
2000	0	1.8	0:56	0:56	1.0	1.8	YES
5000	0	1.8	2:15	2:16	1.0	1.8	YES
10,000	0	1.8	3:41	3:42	1.0	1.8	YES

Table6. NOTIFY- RESPONSE call scenario

OPTION-RESPONSE							
No. of packets	CPU Utilization	Memory Utilization	Time1 (M:S)	Time2 (M:S)	EKIGA Working		
					CPU	Memory	Call
1000	0	1.8	0:21	0:23	1.0	1.8	YES
2000	0	1.8	0:42	0:44	1.0	1.8	YES
5000	0	1.8	1:45	1:46	1.0	1.8	YES
10,000	0	1.8	3:32	3:33	1.0	1.8	YES

Table7. OPTION-RESPONSE call scenario

B. *Quality of VoIP calls under DoS attack*

Here we analyzed the quality of VoIP in term of packet loss, delay and jitter. We analyze the quality under different scenario. First we analyze the quality of call without attack according to figure 5. We find loss of packet and out of order packets are zero for it.

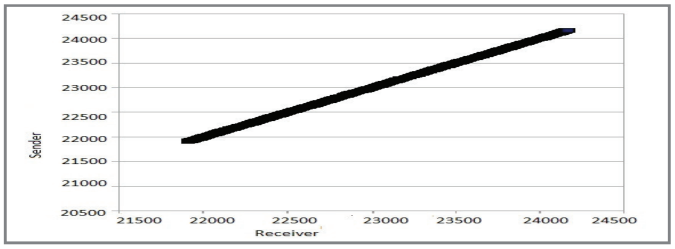

Fig.8. Message sequence of EKIGA to EKIGA call without attack

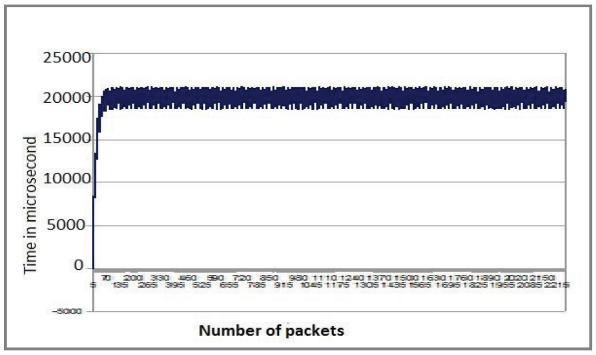

Fig.9. Sender jitter of EKIGA - EKIGA call without attack

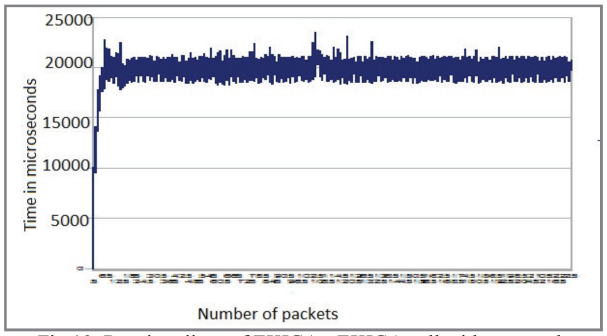

Fig.10. Receiver jitter of EKIGA - EKIGA call without attack

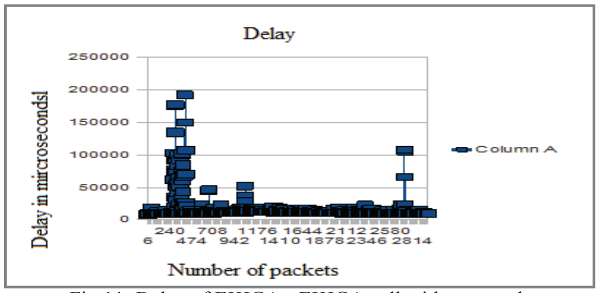

Fig.11. Delay of EKIGA - EKIGA call without attack

At receiver side variation in jitter is very less. Delay is high only for some packets. Figure 8-11 show our results.

Now we analyze the quality of VoIP calls under DoS attack according to figure 6. We analyze the quality of VoIP calls for different scenarios. For every message sequence, we first flood it with 2000 packets, and then stress the server with 100 and subsequently with 200 simultaneous calls. So, here are 3 different scenarios: Server bombarded with only 2000 packets, 2000 packets with 100 simultaneous calls, 2000 packets with 200 simultaneous calls. Then same three tests are conducted for 5000 packets. We repeat these tests for INVITE-BYE call, INVITE-CANCEL call, REGISTER-RESPONSE, MESSAGE-RESPONSE, OPTION–RESPONSE, and NOTIFY-RESPONSE. We have observed that quality of call degrades during this attack. Packets loss and out of order packets remain negligible but delay increases substantially as intensity of flooding and stress on server increases. Here, various graphs depict sender jitter, receiver jitter and delay for various message sequences. Figure 12-23 show result for 2000 packets without any simultaneous call and with 100, 200 simultaneous calls respectively.

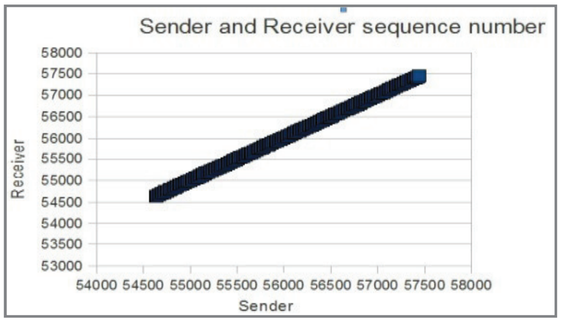

Fig.12. Message sequence of 2000 INVITE-BYE packets Flood

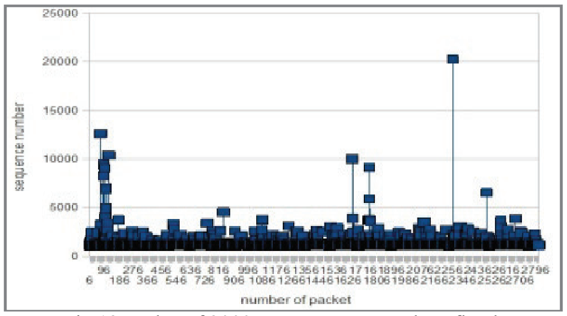

Fig.13. Delay of 2000 INVITE-BYE packets flood

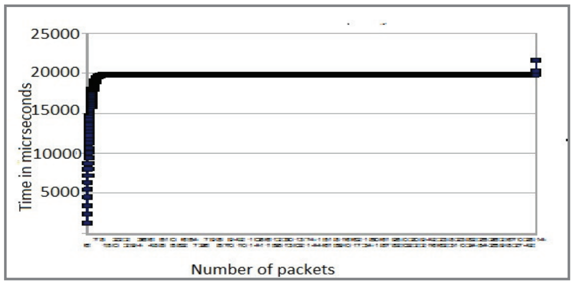

Fig.14. Sender jitter of 2000 INVITE-BYE packets flood

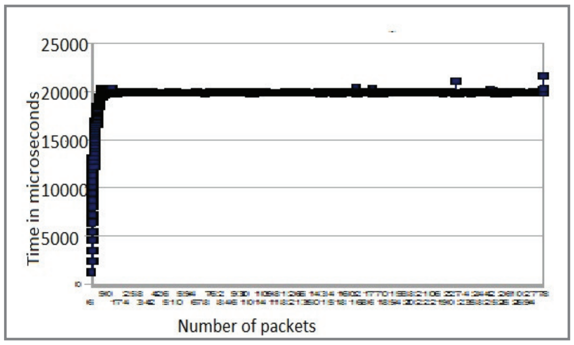

Fig.15. Receiver jitter of 2000 INVITE-BYE packets flood

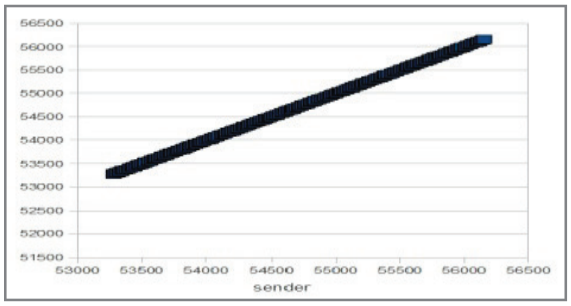

Fig.16. Message sequence of 2000 INVITE packets flood with 100 SIPp calls

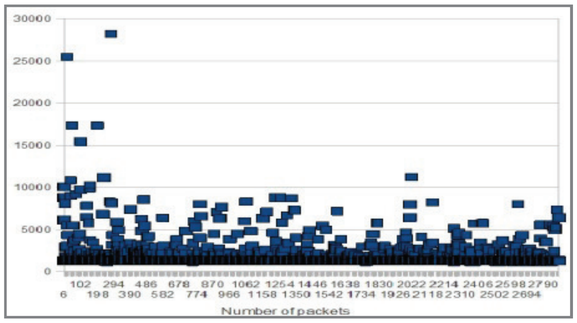

Fig.17. Delay of 2000 INVITE packets with 100 SIPp calls

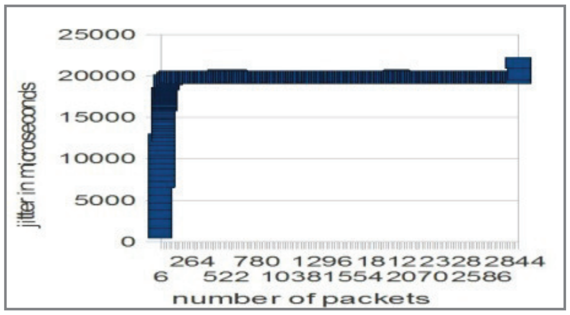

Fig.18. Sender jitter of 2000 INVITEs packets with 100 SIPp calls

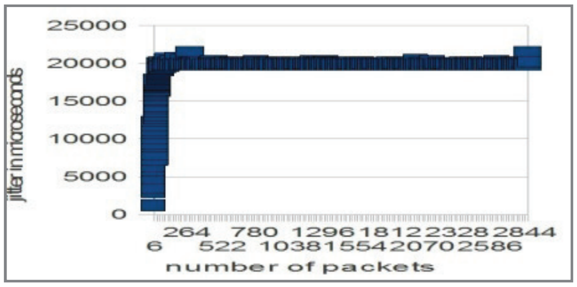

Fig.19. Receiver jitter of 2000 INVITEs packets with 100 SIPp calls

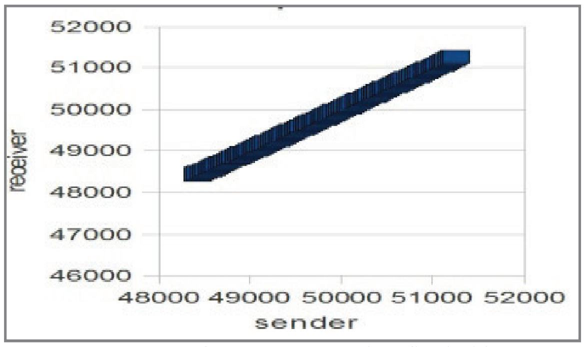

Fig.20. Message sequence of 2000 INVITE packets flood with 200 SIPp calls

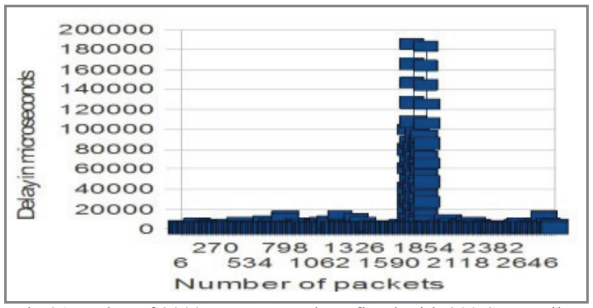

Fig.21. Delay of 2000 INVITE packets flood with 200 SIPp calls

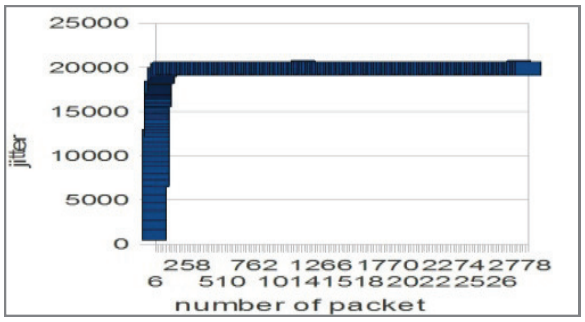

Fig.22. Sender jitter of 2000 INVITE packets flood with 200 SIPp calls

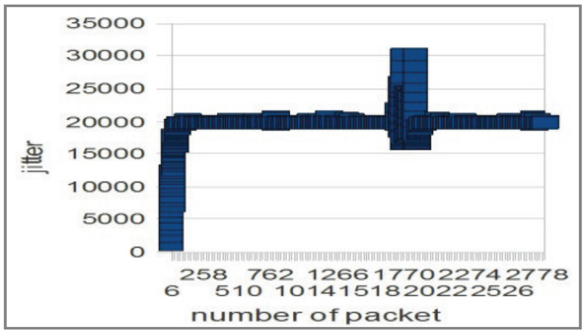

Fig.23. Receiver jitter of 2000 INVITE packets flood with 200 SIPp calls

Figure 11 shows delay without DoS attack and figure 13 shows delay with 2000 packets flood. We can conclude that delay increase during flooding. By figures 11, 13, 17 and 23 we can conclude that during server stress delay increases more.

Figure 10, 15, 19, 22 show receiver jitter without attack, with 2000 packet flood, 2000 packets flood with 100 simultaneous calls, and 2000 packets with 200 simultaneous calls. We conclude that variation at receiver jitter increases as load on server increases. During stress it is more than without stress.

V. Conclusion and Future Work

In this paper, we have evaluated performance of CPU, memory and quality of VoIP calls when SIP server is subjected to bombarding of SIP messages. We have defined the quality metrics for VoIP calls. We have done experiments to answer two questions, raised in Section I:

- Impact of flooding DoS attack on SIP-server.
- Impact of flooding DoS attack on quality of VoIP calls.

We have stressed server with 100-1500 simultaneous calls. We have found that a maximum of 1387 calls could only be made on SIP-server. The quality of VoIP call has been analyzed under flooding of 2000 packets with 100-200 simultaneous calls.

The important observation is that quality of calls goes down significantly in terms of jitter and delay when SIP server is out under stress. Though loss of packets is negligible; excessive flooding of INVITE messages crash the server.

In future, we can extend our work to analyze quality of VoIP calls for more attacks. By doing exhaustive analysis of this result, a method and tool can be devised to mitigate DoS attack.

References

[1] ITU, Draft Revised Recommendation H.323 V5, Geneva, May 2003, pp. 20-30.

[2] Rosenberg J. SIP: Session Initiation Protocol [EB/OL].2002. http://www.ietf.org/rfc/rfc3261.txt.

[3] Rafique, M.Z., Ali Akbar, M. Farooq., "Evaluating DoS Attacks against SIP-based VoIP Systems", Global Telecommunications Conference, 2009. IEEE GLOBECOM 2009. IEEE, pp. 1-6.

[4] Chen, E.Y., "Detecting DoS attacks on SIP Systems", 1st IEEE Workshop on VoIP Management and Security, 2006. pp. 53-58.

[5] Sengar, Haining Wang, "Detecting VoIP Floods Using the Hellinger Distance", IEEE Transactions on Parallel and Distributed Systems, 2008, 19(6):794-805.

[6] D. Sisalem, J. Kuthan, T. Elhert, Denial of Service Attacks Targeting SIP VoIP Infrastructure: Attack Scenarios and prevention Mechanisms. IEEE Network, 2006, pp. 26-31.

[7]M. Voznak and J. Safarik, "DoS Attacks Targeting SIP Server and Improvements of Robustness", International Journal of Mathematics and Computers in Simulation, vol. 6, 2012.

[8] CERT, Denial of Service Attacks, http://www.cert.org/tech_tips/denial_of_service.htm, June 4, 2001

[9] R. Gayraud et al., "SIPp", http://sipp.souceforge.net.

[10] T. Clausen, C. Dearlove, "Jitter", http://tools.ietf.org/html/rfc5148.

[11] Vladimir Tonar, "Jitter", http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Jitter.html.

[12] Asterisk Communication Framework, http://www.asterisk.org.

[13] jNetPcap – open source Protocol Analysis SDK, http://www.jnetpcap.com.