# Fuzzy Logic Overload Control in Asterisk Proxy

Ahmadreza Montazerolghaem
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
Ahmadreza.montazerolghaem@stu-mail.um.ac.ir

Mohammad Hossien Yaghmaee moghaddam
Department of Computer Engineering
Ferdowsi University of Mashhad
Mashhad, Iran
yaghmaee@ieee.org

***Abstract*—The extent and diversity of systems provided by IP networks have lead various technologies to approach integrating various types of access networks and converting to next generation network. On account of features as being in text form, end-to-end connection, independence from the type of transmitted data, and supporting various forms of transmission, is an appropriate choice for signaling protocol in order to make connection between two IP network users. These advantages have made SIP be considered as a signaling protocol in IMS, a proposed signaling platform for next generation networks. Despite having all these advantages, SIP protocol is in lacking of appropriate mechanism for addressing overload. In this paper we try to improve a window-based overload control in RFC 6537. In window-based overload control method, a window is used to limit the number of messages that are sent to an overloaded SIP proxy, simultaneously. In this paper we first use fuzzy logic to regulate the accurate size of window and then we develop, implement, and evaluate it on an Asterisk open-source proxy. Simulation results show that this method could maintain throughput under overload conditions practically, change the maximum window size dynamically.**

***Keywords*— SIP (Session Initiation Protocol), Overload control, Asterisk.**

## I. INTRODUCTION

SIP protocol is the signaling protocol in application layer which is used to start, manage, and finish the meeting between two or more application. The major components of a SIP network are user agents, server proxies, and registrars. User agent is the terminal component in SIP session.

Figure 1 illustrates connection establishment between two user agents in a case in which middle proxies are configured statefully. The proxy task is routing and redeploying signaling between user agents.

SIP server is an application one. The overload problem in SIP server is distinguished with ones in other HTTP servers for at least three reasons: firstly the messages of SIP meeting pass several SIP proxy servers to reach destination which itself could make overload between two SIP proxy servers. Secondly SIP has several retransmit timers which are used for dealing with package loss, especially when the package is sent via UDP transmission protocol, and this could lead to overload on SIP proxy server. Thirdly SIP requests are used as real time session signaling, so have a high sensitivity. Overload in SIP-based networks occur when the server does not have sources necessary for answering every received call. Reviews accomplished in overloaded SIP proxy server show that increasing request rate results in sudden increase in delay in establishing connection and dropping proxy throughput and therefore increase in unsuccessful call rates. Therefore the aim in overload control in SIP is maintaining the throughput of overloaded server near its capacity.

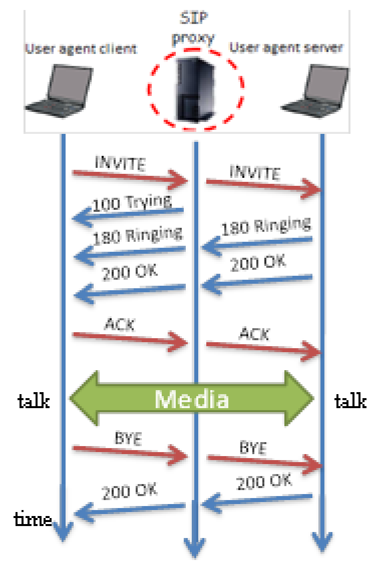


Fig. 1. Exchanged messages for establishing connection in SIP

This study focuses on window based overload control method. The main idea of this method is to limit the number of output messages by controlling the window size. In this paper, we propose fuzzy logic to determine window size as accurately and dynamically as possible, and then we implement and evaluate it on Asterisk open source proxy. Simulation results show that the proposed method reaches a higher throughput than a traditional overload control algorithm proposed in [1].

## II. RELATED WORK

Many researches about the efficiency of SIP proxy server have been accomplished. Paper [2] deals with overload control methods in SIP proxy server and uses OPNET software for measuring throughput. In papers [3] and [4] SIP is practically implemented along with TCP and UDP transmission protocol and OpenSER is used to obtain efficiency results. Articles [5] and [6] mention to window-based distributed method and combination of signal and window-based method, respectively. SIPstone [7] is series of benchmark in which various criterions are proposed for evaluating proxy server powers, redirect

978-1-4799-1228-5/13/$31.00 ©2013 IEEE.

server and registrar in answering SIP requests. In [8] another benchmark is presented for measuring the effect of operating system, hardware configuration, database, and selected transmission layer on SIP efficiency. In [9] practical experiments are accomplished on four types of proxy implementation which are different in both thread management and memory allocation method. The results of these experiments show that the effective parameters in proxy efficiency could be classified in two parts: parameters related to protocol such as message length, length variability, and irregularity of excess load, and parameters related to the type of server implementation e.g. how to allocate sources of operating system to transactions [10].

In [11] various options in selection of transmission layer protocol for SIP are surveyed qualitatively. In [12] the effect of deploying various transmission layer protocols, especially effect of window control mechanism in TCP on throughput and delay in connection establishment is evaluated. In [13] it is shown that despite the general perception in which the more common utilization of UDP than TCP is considered on account of the low processing excess load in the former, it is probable that unfavorable efficiency in TCP utilization is due to implementation manner of proxy.

## III. THE PROPOSED METHOD

In this section we introduce an effective fuzzy-based method for window-based overload control. In this method fuzzy logic is used to solve the problems related to the changes of window size.

In this method a fuzzy controller is contrived to change window size in upstream server dynamically. The input of this controller is the average utilization of CPU and memory in downstream server and its output is the rate of changes of window size in upstream server (ΔW).

In this method the window size control mechanism in upstream proxy is as following:

i. Wmax = Winit

ii. Calculating ΔW by fuzzy controller

iii. $W_{max(t+1)}=W_{max(t)} + (\Delta W * W_{max(t)})$

On the basis of the results of performed experiments, the best range of changes for ΔW is [-0.6, 0.4], which membership function is stated in the following.

In this paper we use Mamdani approach as our fuzzy derivation method.

The proposed fuzzy system in the algorithm includes two input and one output variables.

Membership functions for input and output variables are as shown in the following figures.

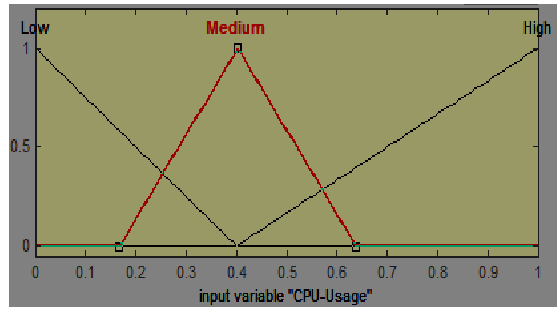


Fig. 2. Membership function of input variable CPU

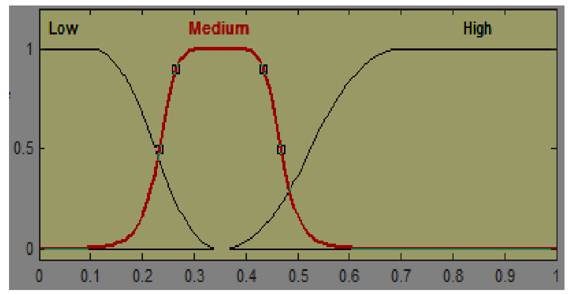


Fig. 3. Membership function of input variable memory

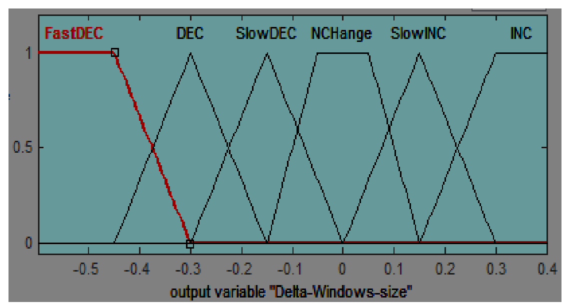


Fig. 4. Membership function of output variable

The rules based used in the proposed algorithm is presented in the following.

TABLE I. FUZZY RULES BASE

| *CPU/memory* | *low* | *Medium* | *high* |
|---|---|---|---|
| *Low* | increase | Slow increase | Slow decrease |
| *Medium* | Slow increase | No Change | decrease |
| *high* | Slow decrease | decrease | Fast decrease |

In order to evaluate rules, first of all inputs are made fuzzy and then applied to the rules premier section. In our system, AND fuzzy operator is used to derive a number representing the assessment of the rules premier section. Then the derived number is applied to the inferior section. Also union operator is used to merge the results of applying fuzzy rules. Consequently a center average de-fuzzier operator is used to derive a real output.

## IV. Network Topologies and Result

The SIP trapezoid [14], shown in figure 5, is used as the basic network topology. In these topology two proxies, namely, the upstream and downstream are used for handling outgoing and incoming calls, respectively. In order to easily study OC performance, the upstream proxy is made faster than the downstream. All calls are originated from the clients of the upstream proxy and are destined to those of the downstream proxy [15].

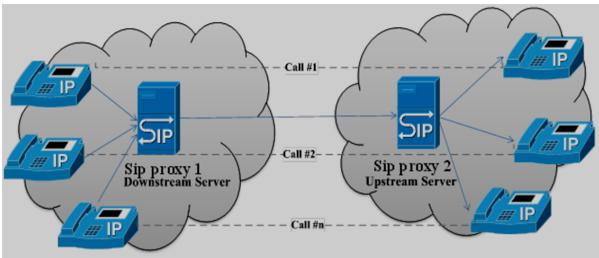


Fig. 5. Dual-Proxy Topology

Asterisk software and Spirent Abacus 5000 tester device are used for implementing proxy servers and user agents, respectively. The upstream server is a PC with INTEL Dual Core 3 GHZ processor and 4 GB memory and the downstream server is a PC with INTEL 1.8 GHz processor and 2 GB memory [16]. Both servers uses version 6.3 of Linux CentOS as their operating system. By modifying Asterisk code we have implemented our proposed mechanism on the upstream server. However the downstream proxy is intact [17]. Also using MATLAB, the proposed fuzzy system is simulated.

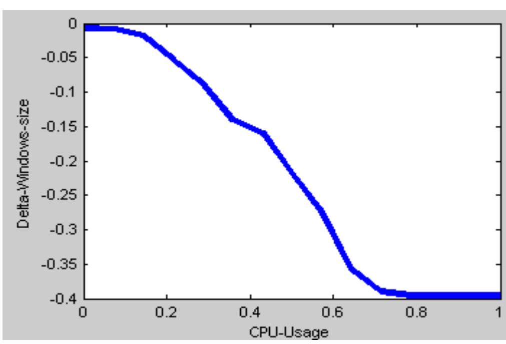


Fig. 6. Windows Variations according to CPU Usage

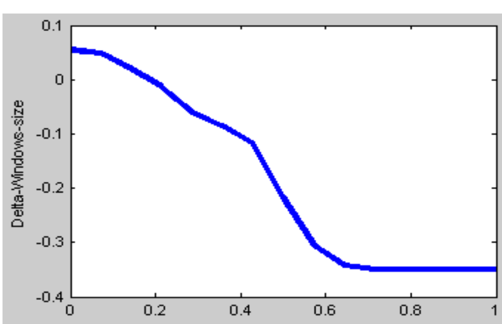


Fig. 7. Windows Variations according to Memory Usage

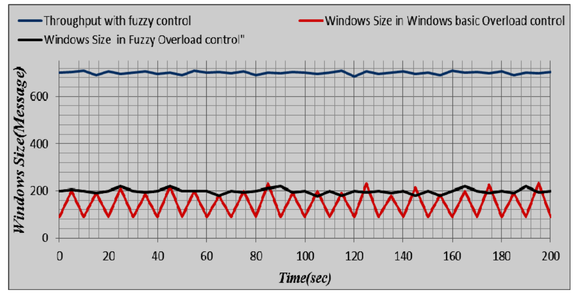


Fig. 8. Windows Variations, throughput with rate 750 cps

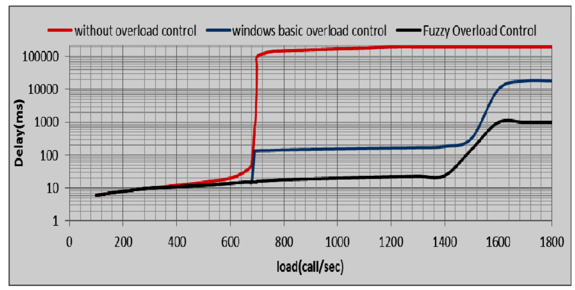


Fig. 9. Average delay comparison of our mechanism with windows basic method and no control

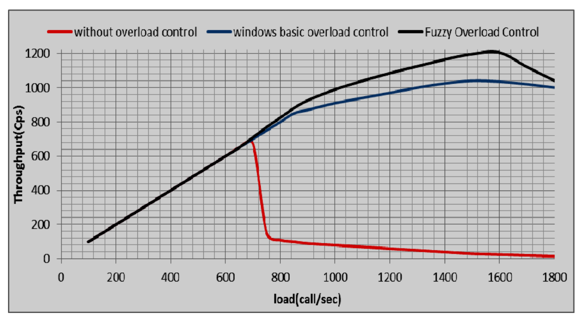


Fig. 10. Performance comparison of our mechanism with windows basic method and no control

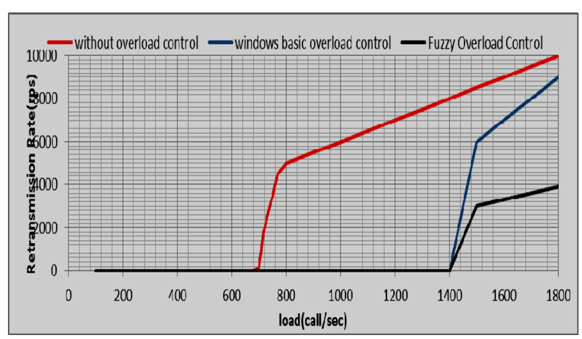


Fig. 11. Retransmission rate comparison of our mechanism with windows basic method and no control

In traditional mechanisms, as new requests are received, window size as well as delay starts to increase. This continuous until delay exceeds a threshold value. At this time window size reduces to its half. Although in fuzzy method, the average window size changes majorly about maximum window size (figure 8).

It is seen that as new requests are received, window size starts to increase and therefore delay increases too. As it is shown in figure 9, this makes increases the average time of call establishment in this proxy to about 1500 cps linearly and with a growth rate far much lower than the case in which the overload control mechanism is not used.

As it is shown in figure 10, using fuzzy overload control mechanism, the upstream proxy is able to maintain its throughput up to about 1500 cps, which is about twice as capacity of downstream proxy, whereas if overload control algorithm is not used, upstream proxy's throughput would be approximately equal to the one related to downstream proxy (700 cps).

The diagram shown in figure 11 illustrates retransmission rate for "INVITE" requests from user side, individually. As it is expected, when we use fuzzy overload control mechanism in upstream server, resend rates of messages decrease considerably.

## V. CONCLUSION

The studies accomplished show that SIP protocol is not efficient enough in facing with congestion, so that when call request rate increases, the delay of call establishment increases suddenly, proxy's throughput falls, and consequently retransmission rates and unsuccessful calls increase. In this paper fuzzy window-based control method is developed, implemented, and tested on a real platform. The suggested method can change the maximum window size dynamically. Studying the charts of throughput, delay, and retransmission rate of "INVITE" shows that our algorithm is able to maintain the throughput at up. As future work, we intend to investigate about more sophisticated window update strategies. In addition, an analytical model as well as stability analysis of the SIP network is also underway.

## ACKNOWLEDGMENT

This work is supported by IP-PBX sample equipment verification laboratory of Ferdowsi University of Mashhad, morally and materially.

## REFERENCES

[1] D. Y. Hwang, J. H. Park, S. W .Yoo, and K. H. Kim, "A window-based overload control considering the number of confirmation Massages for SIP server," presented at the Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on, 2012.

[2] V. Hilt and I. Widjaja, "Controlling overload in networks of SIP servers," presented at the Network Protocols, 2008. ICNP 2008. IEEE International Conference on, 2008.

[3] C. Shen and H. Schulzrinne, "On TCP-based SIP server overload control," presented at the Principles, Systems and Applications of IP Telecommunications, 2010.

[4] E. M. Nahum, J. Tracey, and C. P. Wright, "Evaluating SIP server performance," *SIGMETRICS Perform. Eval. Rev.,* vol. 35, pp. 349-350, 2007.

[5] I. Widjaja, V. Hilt, and H. Schulzrinne, "Session Initiation Protocol (SIP) Overload Control," 2008.

[6] M. Ohta, "Performance comparisons of transport protocols for session initiation protocol signaling," presented at the Telecommunication Networking Workshop on QoS in Multiservice IP Networks, 2008. IT-NEWS 2008. 4th International, 2008.

[7] H. Lindholm, T. Vähäkangas, and K. Raatikainen, "A Control Plane Benchmark for Telecommunications Signalling Applications," *Sort,* vol. 20, p. 100.

[8] S. V. Azhari, M. Homayouni, H. Nemati, J. Enayatizadeh, A. Akbari, " Overload control in SIP networks using no explicit feedback: A window based approach," presented at Computer Communications, Volume 35, Issue 12, 1 July 2012, Pages 1472–1483.

[9] M. Cortes, J. R. Ensor, and J. O. Esteban, "On SIP performance," *Bell Labs Technical Journal,* vol. 9, pp. 155-172, 2004.

[10] S. Wanke, M. Scharf, S. Kiesel, and S. Wahl, "Measurement of the SIP parsing performance in the SIP express router," *Dependable and Adaptable Networks and Services,* pp. 103-110 ,2007.

[11] M. Homayouni, S. V. Azhari, M. Jahanbakhsh, A. Akbari, A. Mansoori, and N. Amani, "Configuration of a sip signaling network: An experimental analysis," presented at the INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on, 2009.

[12] V. K. Gurbani and R. Jain, "Transport protocol considerations for session initiation protocol networks," *Bell Labs Technical Journal,* vol. 9, pp. 83-97, 2004.

[13] K. K. Ram, I. C. Fedeli, A. L. Cox, and S. Rixner, "Explaining the impact of network transport protocols on SIP proxy performance," presented at the Performance Analysis of Systems and software, 2008. ISPASS 2008. IEEE International Symposium on, 2008.

[14] S. S. Gokhale and J. Lu, "Signaling performance of SIP based VoIP: A measurement-based approach," presented at the Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE, 2005.

[15] S. Montagna and M. Pignolo, "Performance evaluation of load control techniques in sip signaling servers," presented at the Systems, 2008. ICONS 08. Third International Conference on, 2008.

[16] P. Montoro, a Comparative Study of VoIP Standards with Asterisk, in: International Conference on Digital Telecommunications, 2009.

[17] D .Qin, Research on the Performance of Asterisk-Based Media Gateway, in: International Symposium on Knowledge Acquisition and Modelling (KAM), 2011.

[18] C. Chi, D. Wang, R. Hao, and W. Zhou, "Performance evaluation of SIP servers," presented at the Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on, 2008.

[19] D. Pesch, M. I. Pous, and G. Foster, "Performance evaluation of SIP-based multimedia services in UMTS," *Computer Networks,* vol. 49, pp. 385-403, 2005.