

Simulation-based Optimization of Signaling Procedures in IP Multimedia Subsystem

Jasmina Barakovic Husic, Alisa Hidic,
Mesud Hadzialic

Faculty of Electrical Engineering, University of Sarajevo
Sarajevo, Bosnia and Herzegovina
{jbarakovic, ah15310, mhadzialic}@etf.unsa.ba

Sabina Barakovic

Ministry of Security of Bosnia and Herzegovina
Sarajevo, Bosnia and Herzegovina
barakovic.sabina@gmail.com

Abstract—This paper presents a simulation-based optimization of signaling procedures in Internet protocol Multimedia Subsystem (IMS). The aim is to improve the performance of signaling procedures by applying an algorithm for Session Initiation Protocol (SIP) message classification and prioritization that is proposed in our previous work. This three-priority level classification of SIP messages is implemented in Network Simulator version 2 (ns-2). Its effectiveness is verified through the simulation-based study of SIP signaling procedures under different conditions. The simulation results are analyzed in terms of Registration Request Delay (RRD), Session Request Delay (SRD), and Session Disconnect Delay (SDD). These SIP performance metrics are improved if IMS is configured to process SIP messages using the proposed algorithm. Differentiated handling of SIP messages reduces the overload in IMS and thereby improves the overall Quality of Service (QoS). This encourages our next step in research activity, i.e., implementation and incorporation of SIP message classification and prioritization algorithm into experimental environment.

I. INTRODUCTION

The Internet protocol Multimedia Subsystem (IMS) supports the development of next generation services. This causes the rapid growth in new services, subscribers and devices that increase the signaling volume. The growing amount of signaling may result in congestion and impact the Quality of Service (QoS) [1]. The IMS procedures used for QoS negotiation and signaling are based on Session Initiation Protocol (SIP). These signaling procedures play an important role in affecting the overall Quality of Experience (QoE) [2]. In this regard, there is a need for an optimization of SIP signaling procedures.

In a large IMS network with millions of potential user agents, several thousand or more SIP messages may be processed by individual SIP proxy servers [3]. Conventional SIP proxy servers are configured to process SIP messages using First In First Out (FIFO) scheduling. However, FIFO scheduling may have several disadvantages: (1) it may be inefficient, (2) it does not

enable service differentiation, (3) it does not permit differentiation based on the classes of service, and (4) it exposes the network to possible instability because of SIP messages that are allowed to loopback through SIP proxy servers.

In the view of these disadvantages, the SIP message classification and prioritization algorithm is proposed in our previous work [4]. Although this algorithm is primarily intended to improve the performance of IMS in high-load and overload conditions, it may be also considered as an extension of our previously published approach [5], which is based on prioritizing of signaling information transmission. According to this approach, signaling service class is given the highest priority over all other user service classes. Although this approach is signaling protocol independent, it is discussed in the context of SIP. Therefore, our previously proposed algorithm further extends this approach by classifying SIP messages into three priority classes.

This paper presents an implementation of this SIP message classification and prioritization algorithm in Network Simulator version 2 (ns-2). The aim is to analyze the impact of implemented algorithm on the SIP performance metrics and compare the results with those obtained using the conventional FIFO scheduling. The intention is to perform the simulation-based optimization of SIP signaling procedures, especially under high-load or overload conditions, in order to improve their performance as an important factor that contributes to the QoE.

The rest of the paper is organized as follows. Section II summarizes the related work. Section III gives an informal and a formal description of algorithm for SIP message classification and prioritization, which is implemented in ns-2. Section IV considers the algorithm's impact on SIP performance metrics. It discusses the obtained results, together with their analysis to show that the conclusions are warranted. Section V concludes the paper and outlines open issues for future work.

II. RELATED WORK

While the performance evaluation of IMS is a subject of emerging research activities, there are many related works regarding the SIP server overload control. Two broad categories of SIP overload control mechanisms are identified in [6]: load balancing approach and load reducing approach. Load balancing approach tries to avoid the overload by distributing the traffic load equally among the local SIP servers. Load reducing approach tries to prevent the overload collapse by reducing the traffic load in the whole SIP network. This approach differentiates three categories of mechanisms for SIP overload control: priority-based, push-back, and retransmission-based. Load balancing mechanisms have been deployed in operator networks, while other three types of load reducing mechanisms are in the stage of research proposals.

The problem of SIP server overload control is the subject of many research activities. As a result, a wide range of mechanisms and algorithms have been developed in order to solve this problem. In this respect, three novel load-balancing mechanisms have been introduced in [7]. Each mechanism combines knowledge of the SIP, dynamic estimates of server load, and Session-Aware Request Assignment (SARA). The proposed mechanisms provide finer-grained load balancing resulting in throughput and response-time improvements.

An optimized algorithm for SIP server overload control that randomly makes the decision of acceptance or rejection of every SIP message is proposed in [8]. This algorithm is based on the calculation of the queue length and the reject probability of SIP messages. The simulation results show that the algorithm better meet the demand on SIP signaling network under overload conditions.

A distributed and adaptive window-based overload control algorithm is proposed in [9]. This algorithm controls the amount of calls that are forwarded to a downstream SIP server in an attempt to prevent it from being overloaded, and does not rely on explicit feedback. This algorithm performs better both in terms of call setup delay and throughput than a commonly used overload control algorithm that is based on maintaining Central Processing Unit (CPU) utilization.

A backpressure-based SIP overload control mechanism called Bassoon is proposed in [10]. It consists of two parts: (1) optimal load balancing algorithm that ensures full utilization of available of available network resources, and (2) end-to-end load control algorithm that regulate traffic at the edge of the network. The Bassoon effectively controls overload in SIP networks and outperforms existing schemes in terms of goodput, fairness and responsiveness.

The implementation and comparison of Adaptive Rate Control (ARC) and Support Vector Machine (SVM)-based

algorithms are described in [11]. The ARC is a queue delay-based algorithm. The SVM-based algorithm is based on integration of four input data: call establishment delay, queuing delay, SIP 100 Trying status code delay, and database response time. The comparison shows that the SVM-based algorithm outperforms ARC algorithm in terms of goodput.

Most of the SIP overload control mechanisms are focused on User Datagram Protocol (UDP), although Transmission Control Protocol (TCP) is more suitable for the transport of SIP messages [12]. Therefore, a novel mechanism that effectively uses TCP flow control to aid application level SIP overload control has been developed. Other experiments indicate that throughput with SIP-over-TCP exhibits similar overload collapse behavior as that with SIP-over-UDP [13].

Priority-based overload control mechanisms aim to mitigate the overload by rejecting the SIP messages with low priority [6]. Prioritization may be performed by using different SIP message header fields [14]. In addition, four approaches to SIP message prioritization are proposed in [15]. A SIP messages scheduling mechanism that is applied on service broker is presented in [16]. This approach balances the load on application servers and enhances the overall QoS. Furthermore, it is proposed an automatic originator regulation of IMS multiple traffic by stateless signal prioritization based on the types, message order and retransmission of SIP messages within each call session [17]. It is shown that best prioritization is to give a higher priority to a message type appearing at a later stage in each session and a lower priority to a retransmitted message. Moreover, light-weight messages like instant messaging are assigned higher priority than those generated by voice calls.

Considering this, the SIP message classification and prioritization algorithm is proposed in our previous work [4]. This algorithm is discussed in detail in Section III.

III. SIP MESSAGE CLASSIFICATION AND PRIORITIZATION ALGORITHM

This paper presents the ns-2 implementation of our algorithm for SIP message classification and prioritization. The decision to use the ns-2 is based on comparative analysis of several simulators [18]. The simulators are compared in terms of modelling capabilities, credibility of simulation models and results, extensibility, usability, and cost of licenses. The analysis shows that commercial OPNET Modeler provides the largest support for IMS simulation [19]. However, ns-2 is chosen because it is free and open-source simulator [20] that provides IMS functionality by adding an independently developed SIP module [21]. This module is based on ns-2 version 2.27.

A. Informal Algorithm Description

The SIP message classification and prioritization algorithm involves two modes of operation: (1) normal mode, wherein the SIP messages are processed using FIFO scheduling, and (2) priority mode, wherein the Priority Queuing (PQ) scheduling is used for three-priority level classification of SIP messages. The normal mode of operation is switched to the priority mode when congestion is detected. The congestion is determined by exceeding the predefined queue length. In priority mode of operation, the packet's content is checked, and packets are classified according to SIP message type.

The class 1 includes SIP messages that terminate the communication session, such as BYE or CANCEL, and those that appear at the later stage in each communication session, such as ACK or 2xx status codes. This class of SIP messages is given the highest priority in the overload conditions. The class 2 includes lightweight SIP messages such as REGISTER, MESSAGE, PUBLISH, NOTIFY, and SUBSCRIBE. These types of SIP messages are less delay sensitive and have lower processing time than INVITE messages [3]. The retransmitted messages are also involved into class 2. A recent studies show that the presence service can account for 50% or more of the total signaling traffic that IMS network handles. Thus, the number of NOTIFY messages is several times larger than the number of other SIP messages. Therefore, this class of SIP messages is assigned medium priority. The class 3 includes the SIP messages that establish the communication session such as INVITE, and those that provide provisional responses, such as 1xx status codes. The class 3 messages are given the lowest priority in the overload conditions [4].

B. Formal Algorithm Description

To implement the previously described algorithm, a modification of existing ns-2.27 source code is done. This modification refers to the adjustment of class *Queue* in order to enable two modes of operations. The modification of class *Queue* entails the modification of class *DropTail*, which implements FIFO scheduling. This class is derived from class *Queue*. The class *Queue* is child class of class *Connector*. This class is derived from class *NsObject*, which is the base class for all network objects in ns-2. This is shown on Fig. 1, which represents a class diagram of the algorithm for SIP message classification and prioritization.

Fig. 2 presents an activity diagram of the proposed algorithm that shows the workflow from a start point to the finish detailing many decision paths that exist in the progression of events contained in the activity. When a class *Queue* receives a packet, the congestion status is checked. The congestion is determined by exceeding the predefined queue length. If there is no congestion, the algorithm operates in normal mode, where packets are served in the same order they have arrived. The class

Queue calls the enqueue function with the packet. If the queue is not blocked, it is allowed to send the packet to its downstream neighbor node. If congestion is detected, the algorithm operates in priority mode. The type of SIP message included into the packet is used to determine the priority value. The packets are served according to the assigned priority value. The high priority value is given to those SIP messages that terminate the communication session, such as the BYE message type. On the other hand, the low priority is assigned to those SIP messages that establish the new communication session, such as the INVITE message type. The medium priority is assigned other types of SIP messages. The formal description of the algorithm's implementation in ns-2 is shown in Pseudocode 1.

IV. SIMULATION-BASED OPTIMIZATION OF IMS SIGNALING PROCEDURES

To verify the effectiveness of SIP message classification and prioritization algorithm, the simulation study is performed.

Pseudocode 1 SIP message classification and prioritization

```

Algorithm SIPMsgClassPrio
receive packet
IF congestion status is true THEN
CASE msgType inside packet OF
CASE reqMethod OF
SM_BYE :
SM_ACK :
SM_CANCEL:
set high priority
SM_INVITE :
set low priority
OTHERS :
set medium priority
ENDCASE
CASE rspCode OF
IF rspCode >= 100 AND rspCode < 200 THEN
set low priority
ELSE IF rspCode >= 200 AND rspCode < 300 THEN
set high priority
ELSE IF rspCode >= 300 AND rspCode < 700 THEN
set medium priority
ELSE
set low priority
ENDIF
ENDCASE
ENDCASE
ELSE
enqueue packet
ENDIF
send packet to node
END SIPMsgClassPrio
    
```

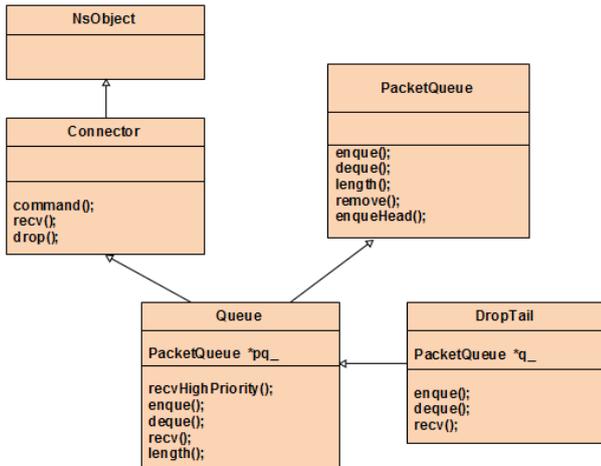


Fig. 1. Class diagram of the algorithm for SIP message classification and prioritization

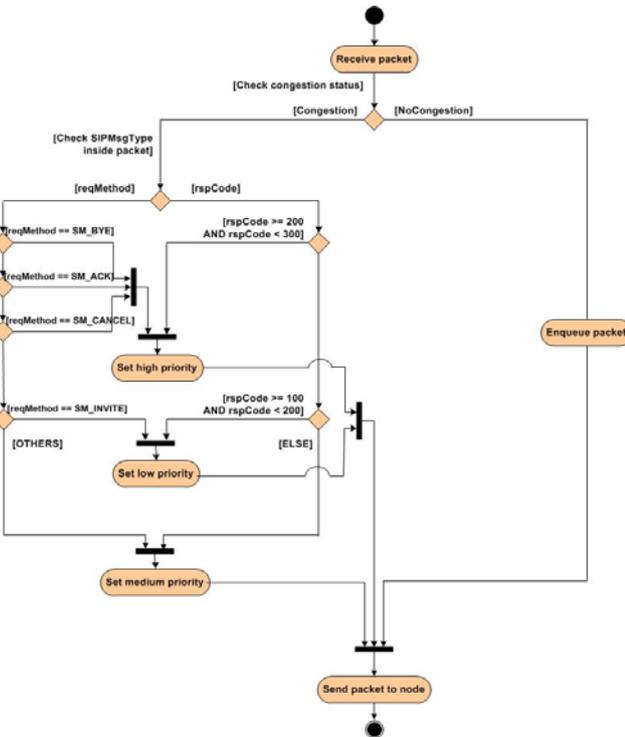


Fig. 2. Activity diagram for SIP message classification and prioritization algorithm

A. Simulation Setup and Environment

The simulations are based on two different scenarios. They differ in the SIP message scheduling algorithm. In Scenario 1, SIP messages are processed using FIFO scheduling. In Scenario 2, SIP messages are processed using proposed algorithm, which is based on FIFO/PQ scheduling.

The simulations are performed on the simple network topology consisting of two boundary routers and one interior router. Boundary routers are connected to SIP domain consisting of SIP proxy servers and SIP user agents. The SIP proxy server incorporates the set of logical functions defined by IMS. It acts as the Call Session Control Function (CSCF) element that proxy SIP signaling between two or more SIP user agents. Due to limitations of used SIP module, it is not possible to simulate other IMS elements. Every SIP domain includes 150 SIP user agents that are used for generating background traffic. The links between routers are dimensioned to implement simple network configuration. The links capacities are configured to 1 Mbps. The delay of all links is set to 10 milliseconds and the queue lengths equal to 100 packets.

The network is loaded by different number of SIP messages generated and exchanged during the signaling procedures. Three types of SIP signaling procedures are considered: registration, session establishment and session termination. Different number of simultaneous SIP signaling procedures is initiated to generate the background traffic. This number is in the range from 0 to 900 for the purpose of measuring the SIP performance metrics. The simulations are run for 500 simulation seconds. Due to nearly permanent characteristics of background traffic, this period can be considered sufficient.

B. Simulation Results and Discussion

The simulation results are analyzed in terms of Registration Request Delay (RRD), Session Request Delay (SRD), and Session Disconnect Delay (SDD). These SIP performance metrics are defined in Request for Comments (RFC) 6076. The simulation results for these SIP performance metrics are shown on Fig. 3. They are not discussed in comparison with results published in related papers because the measurements are performed under different conditions and environment [2], [22]. However, the simulation results are analyzed with the aim to show the impact of the implemented algorithm on considered SIP performance metrics.

Fig. 3(a) shows the comparative analysis of RRD when FIFO and FIFO/PQ scheduling is used. It is noticed that the RRD increases in both scenarios with increasing number of simultaneous SIP signaling procedures. The RRD values are identical in both scenarios until the network is loaded by 60 simultaneous SIP signaling procedures. For a larger number of simultaneous SIP signaling procedures, RRD grows slower when FIFO/PQ scheduling is used. In this case, the RRD is approximately 150 milliseconds lower in overload conditions. This can be explained by that fact that the FIFO is used in normal load conditions, while the PQ is activated when the overload is detected. Moreover, the REGISTER requests are put in class 2 that has a medium priority in high-load and overload conditions.

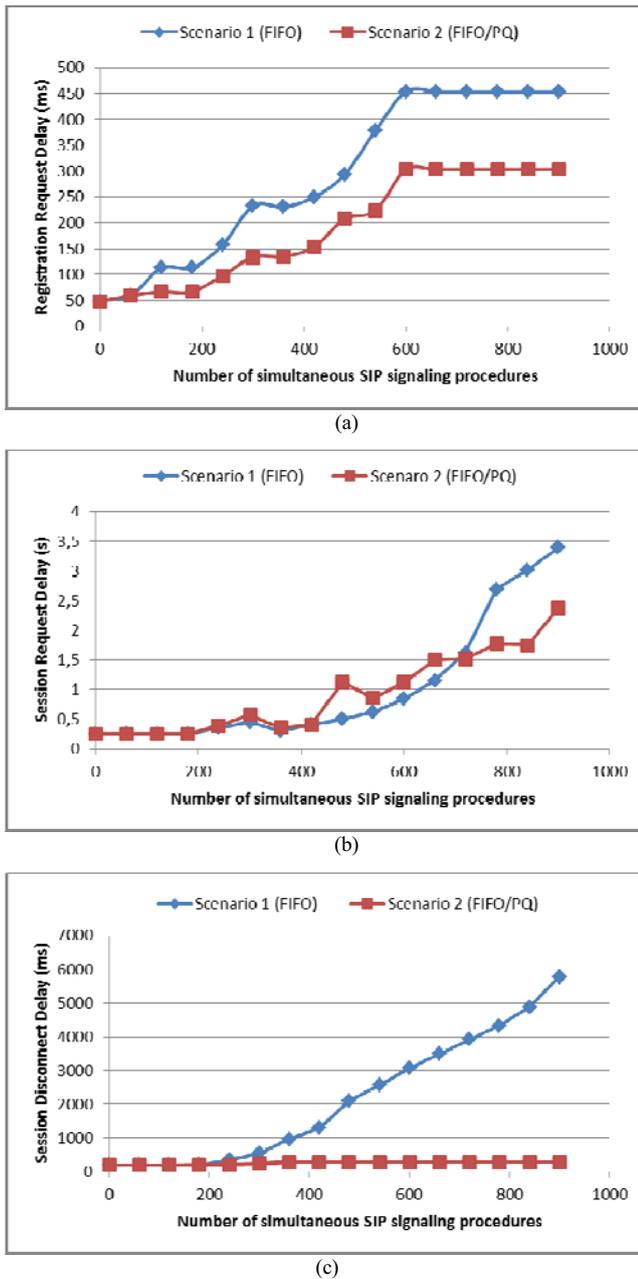


Fig. 1. Simulation results: (a) Registration Request Delay (RRD); (b) Session Request Delay (SRD); (c) Session Disconnect Delay (SDD).

Fig. 3(b) shows the comparison of SRD when FIFO and FIFO/PQ scheduling is used. The SRD values are identical in both scenarios until the network is loaded by 180 simultaneous SIP signaling procedures. The increasing number of SIP signaling procedures leads to the faster deterioration of SRD if FIFO/PQ scheduling is used. This can be explained by the fact that the PQ scheduling is activated in overload conditions giving the lowest priority to INVITE messages. However, when network is loaded by

more than 720 simultaneous SIP signaling procedures, the FIFO/PQ scheduling achieves the better SRD. This is the consequence of giving a higher priority to SIP messages that terminate the existing sessions and thereby reduce the overload. Moreover, this leads to faster processing of INVITE requests and reducing of SRD, consequently.

Fig. 3(c) compares the SDD values obtained when FIFO and FIFO/PQ scheduling are used. The SDD values are identical in both scenarios until the network is loaded by more than 120 simultaneous SIP signaling procedures. This can be explained by the fact that PQ is not activated in normal load conditions. Therefore, better results are achieved in high-load or overload conditions if FIFO/PQ scheduling is used. The insignificant increase of SDD value is noticed until the network is loaded by 480 simultaneous SIP signaling procedures. Then the SDD value becomes constant and equals to 282 milliseconds. This is the consequence of putting the BYE requests into the class 1, which is given the highest priority in high-load and overload conditions.

Summarizing our results, it can be noticed that the proposed algorithm provides nearly the same results as standard FIFO scheduling in normal load conditions. However, it improves the SIP performance metrics in high-load and overload conditions. This is the consequence of two modes of operation: normal mode and priority mode. In normal mode of operation, all SIP messages are processed on the FIFO basis. In priority mode of operation, those SIP messages that terminate session are pushed to the head of the queue by nature of the high priority values. This reduces duration of session termination procedure, which may, for example, improve the billing user experience. On the other hand, assigning a low priority value to SIP messages for session establishment may block a new communication sessions. This may additionally improve the user experience in high-load or overload conditions. This is explained by the fact that users do not accept a service degradation or interruption once they have started a communication session. They would rather have the session to be blocked whenever the network is not able to carry it with the appropriate QoS. Moreover, our three-priority level classification of SIP messages enables the prioritization of different types of service. This may be useful in emergency situations. For example, all SIP messages of one type of service (e.g., instant messaging) may be prioritized over all SIP messages of another type of service (i.e., voice calls). However, this could not be simulated due to the limitations of used SIP module. Therefore, it will be tested in experimental environment during our future research activities.

VII. CONCLUSION

This paper presents the simulation-based optimization of SIP signaling procedures in IMS. The optimization is

achieved by implementing the SIP message classification and prioritization algorithm that is proposed in our previous work. The main idea is to assign a high priority value to SIP messages that may serve to reduce the utilization on the CSCF. Giving a high priority value to SIP messages that terminate existing sessions may reduce the network congestion and thereby improve the overall QoS. Moreover, assigning a low priority value to SIP messages that establish a new session may improve the QoE in case of network congestion.

This SIP message classification and prioritization algorithm is implemented in ns-2 simulator. Its effectiveness is verified through the simulation-based optimization of SIP signaling procedures under different conditions. The simulation results are analyzed in terms of RRD, SRD and SDD. These SIP performance metrics deteriorate with the increasing network load. However, better results are achieved if our algorithm is used instead of conventional approach. Therefore, the simulation results encourage the development and deployment of algorithm for SIP message classification and prioritization in the experimental environment. This is going to be a starting point for our next research activities.

ACKNOWLEDGMENT

This research is supported by the Federal Ministry of Education and Science in Bosnia and Herzegovina under the project "Signaling Information Management to Improve Quality of Next Generation Services".

REFERENCES

- [1] J. Barakovic Husic, S. Barakovic, and H. Bajric, "Evolution of signaling information transmission", *ISRN Commun. and Networking*, vol. 2012, Sept. 2012.
- [2] J. Barakovic Husic, H. Bajric, E. Nekovic, and S. Barakovic, "Basic telephone SIP end-to-end performance metrics", in *Proc. 9th Int. Symp. On Telecommun. (BIHTEL 2012)*, Oct. 2012, pp. 1-6.
- [3] V. S. Abhayawardhana and R. Babbage, "A traffic model for the IP Multimedia Subsystem (IMS)", in *Proc. 65th IEEE Veh. Technology Conf., (VTC Spring 2007)*, Apr. 2007, pp. 783-787.
- [4] J. Barakovic Husic, E. Perenda, M. Hadzialic, and S. Barakovic, "Performance modelling and optimization of IP Multimedia Subsystem", in *Proc. UKSim -AMSS 7th European Modelling Symp. on Math. Modelling and Comput. Simulation (EMS2013)*, Nov. 2013.
- [5] J. Baraković, H. Bajrić, M. Kos, S. Baraković, and A. Husić, "Prioritizing signaling information transmission in next generation network", *J. of Comput. Networks and Commun.*, vol. 2011, Sept. 2011.
- [6] Y. Hong, C. Huang, and J. Yan, "A comparative study of SIP overload control algorithms", in *Network and Traffic Engineering in Emerging Distributed Computing Applications*, IGI Global, 2012, pp. 1-20.
- [7] H. Jiang et al., "Design, implementation, and performance of a load balancer for SIP server clusters", *IEEE/ACM Trans. on Networking*, vol. 20, no. 4, Aug. 2012, pp. 1190-1202.
- [8] J. Yang, F. Huang, and S. Gou, "An optimized algorithm for overload control of SIP signaling network", in *Proc. 5th Int. Conf. on Wireless Commun., Networking and Mobile Computing (WiCom'09)*, Sept. 2009, pp. 1-4.
- [9] M. Homayouni, M. Jahanbakhsh, V. Azhari, and A. Akbari, "Overload control in SIP servers: evaluation and improvement", in *Proc. 17th Int. Conf. on Telecommun.*, Apr. 2010, pp. 666-672.
- [10] Y. Wang, "SIP overload control: A backpressure-based approach", in *Proc. ACM SIGCOMM 2010 Conf.*, Sept. 2010, pp. 399-400.
- [11] Z. Chentouf, "SIP overload control using automatic classification", in *Proc. Saudi Int. Electron., Commun., and Photonics Conf. (SIECPC 2011)*, Apr. 2011, pp. 1-6.
- [12] C. Shen and H. Schulzrinne, "On TCP-based SIP server overload control", in *Proc. Principles, Syst. and Applicat. of IP Telecommun. (IPTComm 2010)*, Aug. 2010, pp. 71-83.
- [13] V. Hilt and I. Widjaja, "Controlling overload in networks of SIP servers", in *Proc. IEEE Int. Conf. on Network Protocols (ICNP 2008)*, Oct. 2008, pp. 83-93.
- [14] A. Acharya, X. Wang, and C. Wright, "A programmable message classification engine for session initiation protocol (SIP)", in *Proc. 3rd ACM/IEEE Symp. on Architecture for Networking and Commun. Syst. (ANCS'07)*, 2007, pp. 185-194.
- [15] M. Cortes, A. Walid, and N. Niethi, "Improving performance of IMS networks with intelligent message processing", in *Proc. Int. Conf. on Intell. Networks (ICIN 2007)*, Oct. 2007.
- [16] Z. S. Khan et al., "Ordering and organization of SIP requests within IMS core by using load awareness and priority", *Int. J. of Recent Trends in Eng.*, vol. 2, 2009, pp. 24-26.
- [17] W. Zhu, A. K. A. Hamid, Y. Kawahara, T. Asami, and Y. Murata, "Automatic originator regulation of IMS multiple traffic by stateless signaling prioritization", in *Proc. Global Commun. Conf. (GLOBECOM 2012)*, Dec. 2012, pp. 2856-2851.
- [18] L. Begg, W. Liu, K. Pawlikowski, S. Perera, and H. Sirisena, "Survey of simulators of next generation networks for studying service availability and resilience", Univ. of Canterbury, Christchurch, New Zealand, Tech. Rep. TR-COSC, Feb. 2006.
- [19] T. Peter, G. Marek, F. Tomas, G. Radoslav, and L. Michal, "Simulation of IMS using current simulators", in *Proc. 50th Int. Symp. ELMAR 2008*, Sept. 2008, pp. 545-548.
- [20] Network Simulator version 2 official website, The ns Manual, Web: <http://www.isi.edu/nsnam/ns/doc/index.html>.
- [21] Rui Prior's website, ns-2 network simulator extension, Web: <http://www.dcc.fc.up.pt/~rprior/ns/>.
- [22] M. Voznak and J. Rozhon, "Methodology for SIP infrastructure performance testing", *WSEAS Transactions on Computers*, vol. 9, no. 9, Sept. 2010, pp. 1012-1021.