ANALYTICAL MODELLING AND SIMULATION FOR PERFORMANCE EVALUATION OF SIP SERVER WITH HYSTERETIC OVERLOAD CONTROL

Konstantin E. Samouylov
Pavel O. Abaev (speaker)
Yuliya V. Gaidamaka
Telecommunication Systems Department
Peoples' Friendship University of Russia
Miklukho-Maklaya str., 6,
117198, Moscow, Russia
Email: {ksam, pabaev, ygaidamaka}@sci.pfu.edu.ru

Alexander V. Pechinkin
Rostislav V. Razumchik
Institute of Informatics Problems of RAS
Vavilova, 44-1,
119333, Moscow, Russia
Email: apechinkin@ipiran.ru,
rrazumchik@ieee.org

KEYWORDS

Signalling network, SIP, hop-by-hop overload control, threshold, hysteretic load control, queuing model.

ABSTRACT

Major standards organizations, ITU, ETSI, and 3GPP have all adopted SIP as a basic signalling protocol for NGN. The current SIP overload control mechanism is unable to prevent congestion collapse and may spread the overload condition throughout the network. In this paper, we investigate one of the implementations of loss based overload control scheme developed by IETF work group which uses hysteretic load control technique on the server side for preventing its overloading. Two different approaches to calculate performance measure of SIP server are introduced. We follow an analytical modelling approach to construct and analyse SIP server model in the form of queuing system with finite buffer occupancy and two-level hysteretic overload control. The formulas for stationary probabilities and the mean return time in the set of normal states were obtained. Simulation is the other approach which allows to eliminate disadvantages of analytical modelling. At present, there is no simulator for modelling of SIP servers in overload conditions with an application of overload control mechanisms which are currently under development by IETF. Approaches to its programming implementations which reflects the protocols and functions that are fully or partially built into the original SIP systems are proposed in the paper.

INTRODUCTION

SIP is an application-layer signaling protocol for creating, modifying, and terminating sessions with one or more participants. In November 2000, SIP was accepted as a 3GPP signaling protocol and main protocol of the IMS architecture. In 2002, recommendation (RFC 3261, 2002) which determines the current protocol form was accepted. The rapid development of the market for services based on the SIP protocol and the growing user needs have revealed a number of shortcomings in the protocol, specifically, in the base overload control mechanism (mechanism 503). In 2009, Rosenberg, one of the protocol designers, demonstrated in (RFC 5390, 2008) the protocols main shortcomings in regard to overload prevention and formulated the main requirements toward the future overload control mechanism. In mid-2010, the SOC working group was created within the IETF Committee. Its work aims at creating overload prevention mechanisms. The first result of their work was the document (IETF draft SIP Rate Control, 2012) permanently accepted in August 2011. The document provides a discussion of the available types of overload control mechanisms local, hop-by-hop, and end-to-end, a classification of SIP networks, and presents the overall architecture of overload-control systems. The SOC group's work focuses on developing two hop-by-hop schemes for overload control as this type of mechanisms has a number of indisputable advantages over the other two types (IETF draft SIP Overload Control, 2013; IETF draft SIP Rate Overload Control, 2013). At present, two overload control schemes have been proposed one with flow sifting on the sender side (LBOC, Loss-based overload control) and one with restricting the flow rate of signaling messages (RBOC, Rate-based overload control). However, only the basic principles were described in SOCs' documents and methods for calculation of the control parameters were not specified. The control parameters can be determined based on analysis of mathematical models or as the results of simulation modeling. As the processes going on in the SIP networks are difficult to describe mathematically and they depend on a large number of different factors, the task needs to be solved through the creation of a simulator.

This paper is organized as follows. In Section 1 we recall SIP client-server model, list of overload control problems that arise because of defect in built-in overload control mechanism, solution requirements that address the problems, and define quality of signaling metrics according to recent standards that fulfill requirements. In Section 2 the hysteretic control mechanism for hop-by-hop overload control based on LBOC scheme is described. In Section 3 we build and analyze a queueing model with threshold control using the embedded

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

Markov chain apparatus. Finally in Section 4 the design and architecture of SIP network simulator is described.

SIP OVERVIEW

SIP is a request/response-based protocol. End users are represented by user agents (UAs), which take the role of a user agent client (UAC) or user agent server (UAS) for a request/response pair. A UAC creates a SIP request and sends it to a UAS. On its way, a SIP request typically traverses one or more SIP proxy servers. The main purpose of a SIP server is to route a request one hop closer to its destination. Responses trace back the path the request has taken.

The UAC sends an INVITE request to the UAS to initiate a SIP session. Each server on the path confirms the reception of this request by returning a 100 Trying response to the previous hop. Instead of forwarding a request, a SIP server can reject it if it is unwilling or unable to forward the request. Once the request is received by the UAS, it typically responds with a 180 Ringing response to indicate that the called user is being alerted and a 200 OK response when the user has accepted the session. After the 200 OK is received by the UAC, it sends an ACK request to complete the three way handshake of an INVITE transaction. The INVITE request is the only SIP request that uses a three way handshake. Sessions can be terminated at any time by sending a BYE request, which is confirmed with a 200 OK response.

SIP Overload Problem and Servers' KPIs

With the increasing deployment of SIP based systems and therefore an increasing number of users that utilize new services, the chance for overload on some nodes rises. A SIP server overload occurs if a SIP server does not have sufficient resources to process all incoming SIP messages. Several reasons, including Poor Capacity Planning, Component Failures, Avalanche Restart or Flash Crowds and the list of the problems that arise as a result of the 503 mechanism are presented in (RFC 5390, 2008), including load amplification problem, underutilization or the Off/On retry-after problem.

Rosenberg formulated 23 requirements to overload control mechanisms in (RFC 5390, 2008); mechanisms matching them will be able to predict and to avoid or quickly to cope with an overload on the server.

It is therefore important to continuously track the current load on all SIP nodes of a telecommunication operator to be able to detect possibly dangerous situations on the one hand as well as to apply load reducing procedures if necessary on the other hand. For this purpose, continuous measurements on respective hosts are required and thus, measurement metrics need to be defined. The IETF has created a working group called IP Performance Metrics (IPPM) and developed a set of metrics that can be applied to the quality, performance and reliability of Internet data delivery services. The base framework for these metrics is defined in RFC 2330 (RFC 2330, 1998). However, this framework is designed for the network layer and is not suitable for application layer protocols. With further progress, the IETF has started another working group in order to define metrics for the remaining layers and named it Performance Metrics for Other Layers (PMOL). This group has standardized its first approach within (RFC 6067, 2010), but, however, the scope of this document is limited to an end-to-end perspective. This perspective does not allow to profile the performance of intermediate entities in the signaling path because it provides only an outside view. Still, it is necessary to define the used measurement metric between two hops in order to detect possible performance problems on a specific intermediate hop. Additionally it is required that the analysis of measurement results based upon these metrics permits a clear differentiation between a overloaded and a non-overloaded system.

The Quality of Signaling metrics, defined in (Happenhofer, 2010) and (Happenhofer2, 2010) have been chosen because they fulfill the mentioned above requirements. The metrics are essential for the following performance analysis, that are:

- Final Response Delay (fRpD) – Time span between sending a request until a final response is received;
- Request Transmits – Total number of requests sent per transaction;
- Success Rate – Ratio of number of successful transactions to injected transactions.

CONCEPT OF HOP-BY-HOP OVERLOAD CONTROL

Current work of the SOC group is focused on the development of two hop-by-hop overload control schemes – Loss-based overload control and Rate-based overload control.

The basic idea of LBOC scheme is that the sending entity (SE) reduces the number of messages on RE's request which will be send to the receiving entity (RE) by specified in the request amount of the total number of messages. RBOC scheme operates in the following way: RE informs SE about the maximum message rate which RE would like to receive from SE within a specified period of time. RE sends the control information to SE periodically depending on RE load changes. Both of these schemes based on the idea of feedback control loop between all neighbouring SIP servers that directly exchange traffic. Each loop controls only two entities. The Actuator is located on the sending entity and throttles the traffic if necessary. The receiving entity has the Monitor which measures the current server load.

The four Via header parameters ('oc', 'oc-algo', 'oc-validity' and 'oc-seq') are introduced in (IETF draft SIP Overload Control, 2013) to transfer the control information between two adjacent entities. The integer parameter 'oc' consisting of 10 digits and its value defines what percentage of the total number of SIP requests are subject to reduction at the SE when the loss-based scheme is used.

Analogously, when the rate-based scheme is used it indicates that the client should send SIP requests at a rate of 'oc'-value SIP requests or fewer per second. 'oc-algo' parameter defines the scope of algorithms supported by SE, e.g. oc-algo="loss", "rate". 'oc-validity' parameter contains a value that indicates an interval of time (measured in milliseconds) that the load reduction specified in the value of the 'oc' parameter should be in effect, its default value is 500 ms. 'oc-sequence' is the sequence number associated with the 'oc' parameter, timestamps usually use as its value.

Threshold Overload Control on the Server Side

As a criteria of determining the choice of moments for sending messages with control information from SE to RE we propose to use hysteretic control technique. The system during operation changes its state depending on the total number of messages n present in it. Choose arbitrary numbers L and H such that $0 < L < H < R$, where R is the buffer capacity. When the system starts to work it is empty, $(n = 0)$, and as long as the total number of messages in the system remains below $H - 1$, system is considered to be in normal state, $(s = 0)$. When total number of messages exceeds $H - 1$ for the first time, the system changes its state to overload, $(s = 1)$, and RE informs SE that traffic load should be reduced: it stays in it as long as the number of messages remains between L and $R - 1$. Being in overload state, RE's system waits till the number of messages drops down below L after which it changes its state back to normal and informs SE about changes, or exceeds $R - 1$ after which it changes its state to blocking, $(s = 2)$, and ask SE for temporary suspension of sending SIP requests. When the total number of messages drops down below $H + 1$, system's state changes back to overload, and RE informs SE that the process of sending of messages can be resumed with the current limitations. Input load function $\lambda(s, n)$ is schematically depicted in Fig. 1.

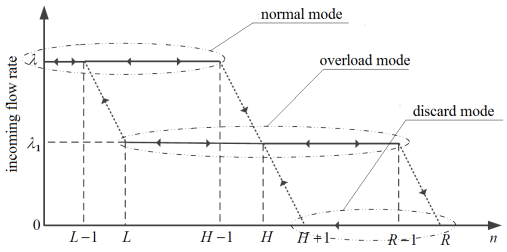

Figure 1: Hysteretic load control

Default Algorithm on the Client Side for LBOC case

In the case of LBOC scheme the default algorithm for throttling incoming to the server traffic is used on the client side. The idea of the algorithm presented in (IETF draft SIP Overload Control, 2013) is to sift the client's outgoing flow. Let us consider the example of the implementation of the algorithm.

The client maintains two types of requests – the priority and non-priority. Prioritization of messages is done in accordance with local policies applicable to each SIP-server. In situations where the client has to sift the outgoing flow, it first reduces non-priority messages, and then if the buffer contains only priority messages and further reduction is still needed, the client reduces the priority messages.

Under overload condition, the client converts the value of the 'oc'=q parameter to a value that it applies to non-priority requests. Let N_1 denote the number of priority messages and N_2 denote the number of the non-priority messages in the client's buffer. The client should reduce the non-priority messages with probability $q_2 = \min\left\{1, q\frac{N_1+N_2}{N_2}\right\}$ and the priority messages with probability $q_1 = \frac{q(N_1+N_2)-q_2N_2}{N_1}$ if necessary to get an overall reduction of the 'oc' value.

To affect the reduction rate with probability q_2 from the non-priority messages, the client draws a random number between 1 and 100 for the request picked from the first category. If the random number is less than or equal to converted value of the "oc" parameter, the request is not forwarded; otherwise the request is forwarded. Recalculation of probabilities is performed periodically every 5-10 seconds by getting the value of the counters N_1 and N_2.

QUEUING MODEL WITH THRESHOLDS CONTROL

Consider a single-queue, single-server system with general service time distribution function which is denoted by $B(x)$ and hysteretic load control. Let denote the Laplace-Stieltjes transform (LST) of $B(x)$ by $\beta(s)$ and the mean service time by $b < \infty$.

Two types of customers (say, type 1 and type 2) arrive at the system in batches (each) in accordance with a Poisson process with rates λ_1 and λ_2 respectively. Henceforth the compound flow rate is denoted by $\lambda = \lambda_1 + \lambda_2$. Each batch has a random number of customers and the probability that arriving batch of type k, $k = 1, 2$, customers contains exactly n, $n \geq 1$, customers is $\omega_{k,n}$.

In further analysis it is assumed that thresholds' values are chosen in such a way that inequalities $H - L \geq 1$ and $R - H \geq 2$ hold.

We turn to the $M^{[X]}|G|1\langle L, H\rangle|\langle H, R\rangle$ queue operating according to the policy that the system may switch between operating modes only at the time instant of a customer departure. If in normal mode just before the customer departure the total number of customers in the system equals H, then the system switches to the overload mode. Similarly if in overload mode just before the customer departure the total number of customers in the system equals H, then the system switches to the discard mode.

Let us denote $X(t)$ a two-dimensional stochastic pro-

cess with set of states

$$S = \left\{ (j,s) \middle| \begin{array}{ll} j = 0,\ldots,R, & s = 0 \\ j = L,\ldots,R, & s = 1 \\ j = H+1,\ldots,R-1, & s = 2 \end{array} \right\}$$

and its subsets $S_i = \{(j,s) \in S | s = i\},\ i = 0,1,2,$ where j is the number of customers in the system and s indicates system operating mode.

Take the service completion epochs to be $0 < t_1 < t_2 < \ldots$, where t_n is the instant of the nth customer departure. Then the discrete-time process embedded at customer departure epochs $X(t_n+0)$ emerges a Markov chain with set of states

$$\tilde{S} = \left\{ (j,s) \middle| \begin{array}{ll} j = 0,\ldots,H-2, & s = 0 \\ j = L,\ldots,R-2, & s = 1 \\ j = H+1,\ldots,R-1, & s = 2 \end{array} \right\};$$

$$\tilde{S}_i = \{(j,s) \in \tilde{S} | s = i\},\ i = 0,1,2.$$

Let us denote $\{p_{j,s}\}$ and $\{q_{j,s}\}$ stationary distribution of $X(t)$ and $X(t_n+0)$ respectively:

$$p_{j,s} = \lim_{t\to\infty} P\{X(t) = (j,s)\},\ (j,s) \in S;$$

$$q_{j,s} = \lim_{n\to\infty} P\{X(t_n+0) = (j,s)\},\ (j,s) \in \tilde{S}.$$

To obtain transition probabilities of the Markov chain we introduce the probability that in operating mode s during the service time of a customer exactly k batches will arrive the system:

$$\beta_k^s = \frac{\lambda_s^k}{k!}\,\beta^{(k)}(\lambda_s),\ s = 0,1,\ k \geq 0.$$

To express transition probabilities of the Markov chain $X(t_n+0)$ we introduce the following auxiliary variables:

$$\omega_i^0 = \delta_i, \omega_i^k = \sum_{n=0}^{i} \omega_{i-n}^{k-1} \frac{\lambda_1\omega_{1,n} + \lambda_2\omega_{2,n}}{\lambda}, k \geq 1, i \geq 0,$$

where δ_i is the Kronecker delta ($\delta_i = 1$ if $i = 0$, or 0 otherwise).

Let us denote by $\alpha_i^s,\ s = 0,1,\ i \geq 0,$ — the probability that in operating mode s exactly i new customers arrive during the time of service of a customer; $A_i^s,\ s = 0,1,\ i \geq 0,$ — the probability that in operating mode s not less than i new customers arrive during the time of service of a customer; $\gamma_i,\ i \geq 0,$ — the probability that immediately after the departure of the customer arrived when the system was empty, there will be exactly i customers in the system:

$$\alpha_i^s = \sum_{k=0}^{i} \beta_k^s \omega_i^k,\ s = 0,1,\ i \geq 0,$$

$$A_i^s = \sum_{k=i}^{\infty} \alpha_k^s,\ s = 0,1,\ i \geq 0,$$

$$\gamma_i = \sum_{k=1}^{i+1} \omega_k^0 \alpha_{i-k+1}^0,\ i \geq 0.$$

Thus the equilibrium equations for probability distribution $\{q_{j,s}\}$ takes the form of

$$\begin{aligned}
q_{j,0} &= q_{0,0}\gamma_j + \sum_{i=1}^{\min(j+1,H-2)} q_{i,0}\alpha_{j-i+1}^0 + \\
&+ \delta_{j-L+1} q_{L,1}\alpha_0^1,\ j = 0,\ldots,H-2, \\
q_{j,1} &= q_{0,0}\gamma_j + \sum_{i=1}^{H-2} q_{i,0}\alpha_{j-i+1}^0 + \\
&+ \sum_{i=L}^{\min(j+1,R-2)} q_{i,1}\alpha_{j-i+1}^1 + \\
&+ \delta_{j-H} q_{H+1,2},\ j = H-1,\ldots,R-2, \qquad (1) \\
q_{j,1} &= \sum_{i=L}^{j+1} q_{i,1}\alpha_{j-i+1}^1,\ j = L,\ldots,H-2, \\
q_{R-1,2} &= q_{0,0} \sum_{i=R-1}^{\infty} \gamma_i + \\
&+ \sum_{i=1}^{H-2} q_{i,0} A_{R-i}^k + \sum_{i=L}^{R-2} q_{i,1} A_{R-i}^1, \\
q_{j,2} &= q_{R-1,2},\ j = H+1,\ldots,R-2.
\end{aligned}$$

The probability $q_{0,0}$ is determined from the normalization condition.

Stationary state distribution

We use the renewal theory to receive the stationary queue length distribution of the corresponding stochastic process from the stationary queue length distribution of the embedded Markov chain.

The stationary mean T of the time interval between neighboring instants t_n and t_{n+1} is defined by the formula

$$T = b + \frac{1}{\lambda_0} q_{0,0}.$$

We also denote $\nu = 1/T$,

$$\tilde{\beta}_k^s = \frac{\lambda_s^k}{k!}\tilde{\beta}^{(k)}(\lambda_s),\ s = 0,1,\ k \geq 0,$$

$$\tilde{\alpha}_i^s = \sum_{k=0}^{i} \tilde{\beta}_k^s \omega_i^k,\ s = 0,1,\ i \geq 0,$$

$$\tilde{A}_i^s = \sum_{k=i}^{\infty} \tilde{\alpha}_k^s,\ s = 0,1,\ i \geq 0.$$

$$\tilde{\gamma}_i = \sum_{k=1}^{i} \omega_k^0 \tilde{\alpha}_{i-k}^0,,\ i \geq 1.$$

The following theorem contains formulas for calculating the stationary distribution $\{p_{j,s}\}$.

Theorem. *Stationary probabilities of the stochastic process $X(t)$ are given by*

$$p_{0,0} = \frac{\nu}{\lambda_0} q_{0,0},$$

$$p_{j,0} = \nu \left(\tilde{\gamma}_j q_{0,0} + \sum_{i=1}^{\min(j,H-2)} \tilde{\alpha}^0_{j-i} q_{i,0} \right), \quad j = 1, \ldots, R-1,$$

$$p_{R,0} = \nu \left(\sum_{i=R}^{\infty} \tilde{\gamma}_i q_{0,0} + \sum_{i=1}^{H-2} \tilde{A}^0_{j-i} q_{i,0} \right),$$

$$p_{j,1} = \nu \sum_{i=L}^{\min(j,R-2)} \tilde{\alpha}^1_{j-i} q_{i,1}, \quad j = L, \ldots, R-1,$$

$$p_{R,1} = \nu \sum_{i=L}^{R-2} \tilde{A}^1_{j-i} q_{i,1},$$

$$p_{j,2} = \nu b q_{R-1,2}, \quad j = H+1, \ldots, R-1.$$

Performance measures

We denote by $P(S_1) = \sum_{j=L}^{R} p_{j,1}$ the stationary probability of the system being in overload mode, by $P(S_2) = \sum_{j=H+1}^{R-1} p_{j,2}$ the system being in discard mode, by $P(S_0) = \sum_{j=0}^{H-2} p_{j,0}$ the system being in normal mode.

The mean control cycle time is inverse to stationary intensity of instants of control cycle starts. Since the control cycle starts when the system passes from state $(L,1)$ to state $(L-1,0)$, then the stationary intensity of instants of control cycle starts is equal to the stationary intensity of passes from state $(L,1)$ to state $(L-1,0)$ is defined as follows:

$$\mu = \nu q_{L,1} \int_0^{\infty} e^{-\lambda_1 x} dB(x) = \nu \beta^1_0 q_{L,1} = \tau^{-1}.$$

Thus mean time τ_{12} the system spends in overload and discard set of states during one control cycle can be calculated by the following formula

$$\tau_{12} = \frac{P(S_1) + P(S_2)}{P(S_0) + P(S_1) + P(S_2)} \cdot \frac{1}{\nu \beta^1_0 q_{L,1}}.$$

Numerical Example

Below we provide a numerical example relating to the mean return time of the system from the overload mode to the normal load mode in the case of small dimension, i.e. with the values of the thresholds $L = 8$, $H = 12$ and $R = 20$. We consider two versions of the service time distribution: an exponential service time and constant service time with the same mean value of $b = 1$. Let the distribution of the number of customers in the batch be $\omega_{k,n} = 0.2$, where $k = 1, 2$ and $n = \overline{1,5}$. Let the intensity of the input flow be $\lambda = 2/3$, and since the average number of customers in the batch is equal to 3, then the offered load intensity takes a value $\rho = 2$.

Fig. 2 presents the results of the calculations in the form of graphs showing the dependence of mean return time τ_{12} on the dropping probability q for the two policies of overload control. The mean service time is taken per time unit. One can note that the mean return time for an exponential service time is less than those for a constant time for the values of $q < 0.6$

SIP NETWORK SIMULATION

The network testbed that we used to generate calls and collect measurement data consists of UAC, UAS and SIP proxy. The instance of SIPp tool modified to support Loss based overload control scheme was run on UAC and UAS. On the server side the SIP server instatance that supported basic FSM for succesfull call setup scenario was run.

Figure 3 illustrates the steps a packet takes as it moves from the device driver through the Linux kernel to the SIP layer and back to the device driver, and then we introduce the methodology we used to measure all the components of the packet service and waiting times.

Based on this figure, there are three distinct entities involved in processing a SIP packet. The first one of them is Kernel Network Stack, which provides the procedure of receiving of packets. As soon as a packet is received from the physical device, it arrives at the device driver and is transferred to a ring buffer in kernel space. The packet then undergoes processing within the Linux kernel stack before it is handed to the application – SIP layer.

Application Layer provides SIP Packet Processing procedures. The SIP layer has a blocking loop waiting for packets to arrive on the socket. Processing at the SIP layer consists of two parts: one that is common to all messages and one that is message dependant. During the common part, the message is parsed, it is classified as a SIP request or response, and then certain operations are performed. Since the proxy operates in stateful mode, a lookup is performed to determine if the SIP transaction is already present. If not, a new transaction is created, otherwise the existing transaction is returned and the SIP message-specific processing phase begins. Once the forwarding, reply or relay decision has been made, a send buffer is created with the updated data for the SIP

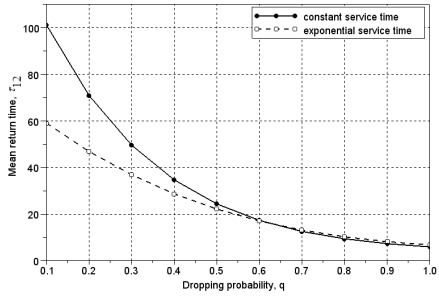

Figure 2: Dependence of mean return time τ_{12} from dropping probability q

header, and the packet is sent to the transport layer. The next layer Kernel Network Stack provides Packet Sending procedures. Once a packet completes processing at the SIP layer, it is passed on to the kernel for forwarding to the UAC/UAS.

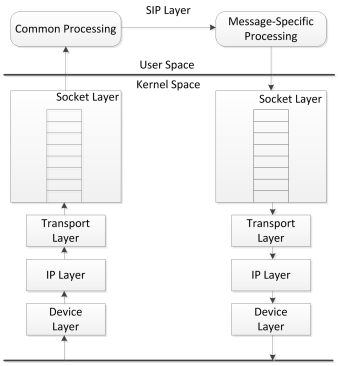

Figure 3: Message flow path through OS and SIP layers

First, for parsing, the text based SIP messages are syntactically analysed, broken down into parts, and converted into internal representations. Second, the newly created internal message representations have to be analyzed to infer on their later processing. For a SIP proxy server this means that it has to determine the messages destination and forward them towards there. For a UAS it means extracting the content of the message, probably displaying something to the user and creating and sending responses. Each of the two parts of processing a SIP message in a proxy server uses a specific amount of processing time. We assumed that parsing of a message always takes approximately the same amount of processing time, which independent of its type and content. The simulation model distinguishes requests and responses and assumes further that a response needs less processing time than a request. These assumptions lead to the proxy simulation model depicted in Fig. 4. Within this model, each message type refers to one of the four specific queues which are served by the processor P, according to priority scheduling (that means that a message from the FCFS queue with the highest priority is always taken first and forwarded to the processor; if there are no messages in the first queue left, a message with the next lower priority is taken, and so on). The following priorities are given to the message types (in decreasing order): Incoming (unparsed) external messages, Self-created retransmissions, Parsed responses, Parsed requests.

A queueing Model (upper part) and a transaction manager (lower part) were added to the description above as main components. Upon the processing of a message, the SIP-specific transaction manager creates or deals with the corresponding state machine and returns the message into the queuing model for a second time, either as a retransmission, request or response message.

Note that we assume that message reception and parsing is of highest priority, as it leads to the creation of the basic internal message representations in the memory. This model is based on an interrupt-driven system and therefore the network interface card triggers an interrupt upon reception of a message. That means further, the normal operation (that is SIP routing in terms of the simulation model) is interrupted as often as a new SIP message is received. After creating these structures, each message is again queued for routing until it is processed (routed). Within the routing process, the simulator creates transactions for each new request and they set timers which possibly create retransmissions of requests which will then be sorted into the re-transmission queue.

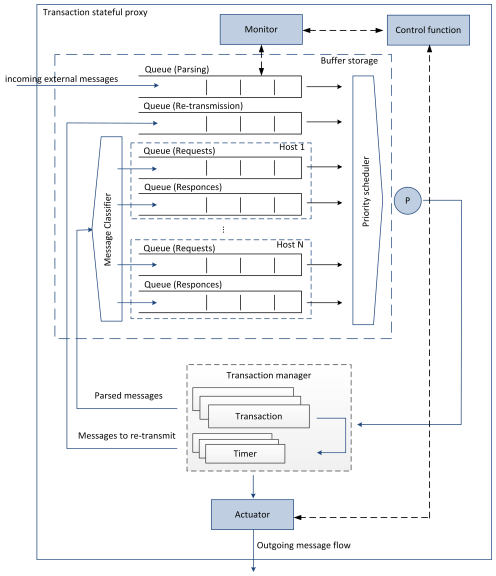

Figure 4: Message scheduling in proxy

CONCLUSION

In this paper we have considered two approaches for estimation performance measures of SIP server which operates in accordance with hysteretic overload control algorithm. We have built and analysed a queueing model with threshold control using the embedded Markov chain apparatus. The obtained formulas for calculation of the mean return time to the set of normal states are focused on numerical implementation. We have also presented the SIP simulator and considered several KPIs for its performance estimation.

Future work will be focused on mathematical modeling of overload control algorithms to make a proposal for the preliminary values of the control parameters, which will be used as input data for the simulator, and on investigation the impact of control parameters on the effectiveness of server performance. Development of the simulator will be continued to support the possibility of KPI's calculation and expansion of implemented FSM that includes different SIP call flow scenarios.

Notes and Comments. This work was supported in part by the Russian Foundation for Basic Research (grants 12-07-00108 and 13-07-00665).

REFERENCES

Abaev, P., Gaidamaka, Yu., Samouylov, K. 2012. Modeling of Hysteretic Signaling Load Control in Next Generation Networks. Lecture Notes in Computer Science. Germany, Heidelberg, Springer-Verlag. –Vol. 7469. –P.371–378.

Abaev, P., Gaidamaka, Yu., Samouylov, K. 2012. Queuing Model for Loss-Based Overload Control in a SIP Server Using a Hysteretic Technique. Lecture Notes in Computer Science. Germany, Heidelberg, Springer-Verlag. -Vol. 7469. –P.440–452.

Abaev, P., Gaidamaka, Yu., Pechinkin, V., Razumchik, R., Shorgin, S. 2012. Simulation of overload control in SIP server networks. Proceedings of the 26th European Conference on Modelling and Simulation, ECMS 2012. –Germany, Koblenz. –Pp. 533–539.

Abaev, P., Gaidamaka, Yu., Samouylov, K., Shorgin, S. 2013. Design And Software Architecture Of Sip Server For Overload Control Simulation. Proceedings of the 27th European Conference on Modelling and Simulation, ECMS 2013. –Norway, Alesund. –Pp. 580–586.

Abaev, P., Pechinkin, V., Razumchik, R. 2012. On analytical model for optimal SIP server hop-by-hop overload control. Communications in Computer and Information Science: Modern Probabilistic Methods for Analysis of Telecommunication Networks. Germany, Heidelberg, Springer–Verlag. –Vol. 356. –P.1–10.

Rosenberg, J., Schulzrinne, H., Camarillo, G. et al. 2002. SIP: Session Initiation Protocol. RFC 3261.

Gurbani, V., Hilt, V., Schulzrinne, H. 2013. Session Initiation Protocol (SIP) Overload Control. draft-ietf-soc-overload-control-14.

Hilt, V., Noel, E., Shen, C., Abdelal, A. 2011. Design Considerations for Session Initiation Protocol (SIP) Overload Control. RFC 6357.

Noel, E., Williams, P. 2012. Session Initiation Protocol (SIP) Rate Control. draft-ietf-soc-overload-rate-control-03.

Rosenberg, J. 2008. Requirements for Management of Overload in the Session Initiation Protocol. RFC 5390.

Rosenberg, J. 2008. Session Initiation Protocol (SIP) Basic Call Flow Examples. RFC 3665.

Gaidamaka, Yu., Pechinkin, A., Razumchik, R., Samouylov, K., Sopin, E. 2014. Analysis of M/G/1/R Queue with Batch Arrivals and Two Hysteretic Overload Control Policies. International Journal of Applied Mathematics and Computer Science (in print).

Paxson, V., Almes, G., Mahdavi, J., Mathis, M. 1998. Framework for IP Performance Metrics. RFC 2330.

Davis, M., Phillips, A., Umaoka, Y. 2010. BCP 47 Extension U. RFC 6067.

Happenhofer, M., Egger, C., Reichl, P. 2010. Quality of Signaling: A new Concept for Evaluating the Performance of Non-INVITE SIP Transactions. Proc. of 22nd International Teletraffic Congress (ITC), 2010.

Happenhofer, M., Reichl, P. 2010. Quality of Signaling (QoSg) Metrics for Evaluating SIP Transaction Performance. In The 18th International Conference on Software, Telecommunications and Computer Networks. —Pp. 270-274.

AUTHOR BIOGRAPHIES

KONSTANTIN E. SAMOUYLOV received his Ph.D. from the Moscow State University and a Doctor of Sciences degree from the Moscow Technical University of Communications and Informatics. During 1985–1996 he held several positions at the Faculty of Sciences of the Peoples' Friendship University of Russia where he became a head of the Telecommunication Systems Department in 1996. His current research interests are performance analysis of 3G networks, teletraffic of triple play networks, and signaling networks planning. He is the author of more than 100 scientific and technical papers and three books. His email address is ksam@sci.pfu.edu.ru.

PAVEL O. ABAEV received his Ph.D. in Computer Science from the Peoples' Friendship University of Russia in 2011. He is an Associate Professor in the Telecommunication Systems department at Peoples Friendship University of Russia since 2013. His current research focus is on NGN signalling, QoS analysis of SIP, and mathematical modeling of communication networks. His email address is pabaev@sci.pfu.edu.ru.

YULIYA V. GAIDAMAKA received the Ph.D. in Mathematics from the Peoples' Friendship University of Russia in 2001. Since then, she has been an associate professor in the university's Telecommunication Systems department. She is the author of more than 50 scientific and conference papers. Her research interests include SIP signalling, multiservice and P2P networks performance analysis, and OFDMA based networks. Her email address is ygaidamaka@sci.pfu.edu.ru.

ALEXANDER V. PECHINKIN is a Doctor of Sciences in Physics and Mathematics and principal scientist at the Institute of Informatics Problems of the Russian Academy of Sciences, and a professor at the Peoples' Friendship University of Russia. He is the author of more than 150 papers in the field of applied probability theory. His email address is apechinkin@ipiran.ru.

ROSTISLAV V. RAZUMCHIK received his Ph.D. in Physics and Mathematics in 2011. Since then, he has worked as a senior researcher at the Institute of Informatics Problems of the Russian Academy of Sciences. His current research activities focus on stochastic processes and queuing theory. His email address is rrazumchik@ieee.org