Computing and Informatics, Vol. 33, 2014, 218–236

HYSTERETIC CONTROL TECHNIQUE FOR OVERLOAD PROBLEM SOLUTION IN NETWORK OF SIP SERVERS

Pavel Abaev, Yulia Gaidamaka, Konstantin Samouylov

Telecommunication Systems Department
Peoples' Friendship University of Russia
Miklukho-Maklaya str. 6
117198 Moscow, Russia
e-mail: {pabaev, ygaidamaka, ksam}@sci.pfu.edu.ru

Alexander Pechinkin, Rostislav Razumchik, Sergey Shorgin

Institute of Informatics Problems
Russian Academy of Sciences
Vavilova str. 44-2
119333 Moscow, Russia
e-mail: apechinkin@ipiran.ru, rrazumchik@ieee.org, sshorgin@ipiran.ru

Abstract. This paper contains research and development results concerning application of hysteretic control principles to solve SIP servers overload problem, which is known from a number of IETF standards and scientific papers published over the past few years. The problem is that SIP protocol, being the application layer protocol, by default has no build-in means of overload control, as, for example, SS7, MTP2 and MTP3 protocols. It was the SS7 network, where a threshold mechanism of hysteretic signaling load control was first implemented. In this paper we describe the main up-to-date solutions of an overload control problem in a signaling network, and develop analytical models of hysteretic control, which are useful in the development of load management functions of SIP servers. We also propose the design of Open SIP signaling Node (OSN) software architecture which is intended to be used for simulations and comparison of various overload control mechanisms.

Keywords: Signaling network, SIP servers network, hop-by-hop overload control, threshold, hysteretic load control, load-based overload control, queuing model, simulator architecture

1 INTRODUCTION

We investigate threshold hysteretic control technique which may be able to solve the problem of overload control in a signaling network, built on the SIP-based servers [3, 4]. This problem is known from the IETF documents [5, 6, 7], and from a number of scientific papers [8, 9, 10, 11, 12, 13, 18, 19, 20]. Particular attention is paid to solving the problem of hysteresis control technique for loss-based overload control (LBOC) scheme, proposed in the document IETF [6]. During the development of hysteretic overload control mechanism we applied the solution designed for SS7 networks [1, 2] and analytical models in the form of queuing systems, developed and analyzed by the authors in [18, 19, 20]. We also propose architecture of the Open SIP signaling Node (OSN) software for analysing QoS parameters of SIP server signaling network.

The problem with SIP is at the application layer of OSI/ISO model; the protocol has no built-in mechanisms of protection against overloads. This turned out to be the key factor that led to a series of research activities outlined in RFC 5390 [4], devoted to analysis of SIP servers overload. Recently, in RFC 6357 [5] several methods for overload control have been proposed, among which the most developed scheme is LBOC. However, even in this case, the use of specific control mechanisms is up to the developers, in contrast to the SS7, where all the control mechanisms are built into MTP2 and MTP3 protocols, and adopted as an international standard. It is worth noticing that one of the possible solutions of overload control problem in next generation networks may be the same as in SS7, i.e. it may be solved by creating a dedicated common session signaling (CSS) network.

This paper is organised as follows. First, we analyse the previous experience of international standards organizations for SS7 and SIP-signaling overload control problem solutions. Second, we formulate the problem of hysteretic load control based on the experience of SS7. Third, in accordance with the conclusions drawn in the previous section, we construct an analytical model of LBOC scheme in the form of a queuing system, and we describe the method of analysis of the key control parameter - the return time from the overload states of the system to the normal load set of states. Finally, we introduce the OSN software architecture with LBOC scheme implemented.

2 FORMULATION OF THE PROBLEM OF HYSTERETIC LOAD CONTROL APPLICATION

SS7 signaling link congestion control developed by the ITU is based on the technique of hysteretic load control. One of the first results in this area was published in [16]. The technique implements monitoring of the total number of messages in transmit or retransmit buffer and signaling traffic flow control. The signaling traffic flow control procedures are used to divert a given traffic flow (toward one or more destinations) from congested signaling link to an alternative available signaling link.

The mechanism of SS7 signaling link congestion control consists of two stages: to detect congestion and to eliminate or mitigate congestion.

In order to detect congestion the monitoring of the total number of messages in transmit and retransmit buffers is kept on hold. To eliminate congestion, limitation of signaling traffic at its source is performed. Limitation (restriction or prohibition) of incoming signalling traffic is needed in case the signalling network is not capable of transferring all signalling traffic offered by the user because of network failures or congestion. The following three types of thresholds are defined in ITU Recommendation Q.704:

- congestion onset threshold H_1 – for detecting the onset of congestion,
- congestion abatement threshold L_1 – for monitoring the abatement of congestion,
- congestion discard threshold R_1 – for determining whether, under congestion conditions, a message should be discarded.

In national signalling networks with multiple congestion thresholds up to three separate thresholds are provided for detecting the onset of congestion, and up to three separate thresholds for monitoring the abatement of congestion, respectively. In addition, up to two separate thresholds are provided for determining whether, under congestion conditions, a message should be discarded or transmitted using the signalling link (Figure 1). Let us point out that Figure 1 shows the case $R_i < L_{i+1}$, but the case $R_i > L_{i+1}$ is not considered in this paper.

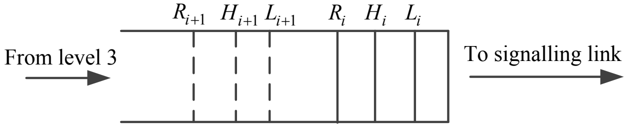


Figure 1. Signalling link congestion status thresholds

So there are four congestion detection statuses h and three congestion discard statuses r:

$$h = \begin{cases} 0, & \text{no congestion,} \\ i, & i - \text{level of congestion}, i = 1, 2, 3; \end{cases}$$

$$r = \begin{cases} 0, & \text{no discard,} \\ i, & i - \text{level of discard}, i = 1, 2. \end{cases}$$

Criteria for determination of signalling link congestion status are the number of messages in transmit or retransmit buffers, i.e. buffer occupancy. When the buffer occupancy increases and exceeds congestion onset threshold, H_1, congestion is determined. Then the incoming load is reduced to avoid overloading. However, the load does not return to normal load value immediately, but does so after a while when the buffer occupancy decreases and comes to below the congestion abatement threshold, L_1. This technique is called hysteretic load control. Two thresholds (onset and abatement) are needed to reduce potential oscillations between control-on and control-off states under certain loading conditions.

Under normal load conditions when the signalling link is uncongested, the signalling link congestion status is assigned zero value ($h = 0$). When the buffer occupancy increases the congestion status does not change until the predetermined congestion onset threshold, H_1, of the buffer occupancy is crossed. After that the current congestion status is assigned the unit value ($h = 1$). When the buffer occupancy is increasing up to the congestion discard threshold, R_1, or when the buffer occupancy is decreasing up to the congestion abatement threshold L_1, the congestion status has the unit value ($h = 1$). When the buffer occupancy is decreasing and crosses the congestion abatement threshold, L_1, the congestion status is assigned zero value ($h = 0$).

Figure 2 shows hysteretic load control where the incoming signalling load $\lambda(h, r, n)$ depends on congestion detection and discard statuses, and buffer occupancy.

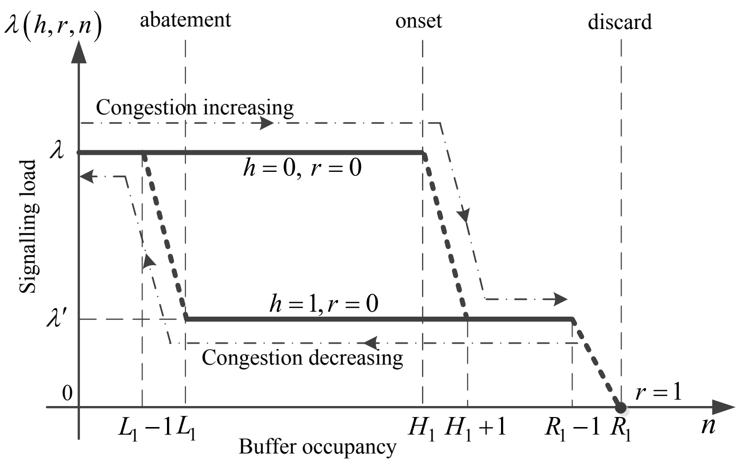


Figure 2. Signalling load hysteretic control

Under normal load conditions the signalling traffic load has the value λ. When congestion is detected after the buffer occupancy has crossed the congestion onset threshold H_1, the normal signalling traffic load is restricted by the value, λ'. When the buffer occupancy is increasing and crosses the congestion discard threshold R_1, the signalling traffic load is prohibited, i.e. $\lambda(h, r, n)$=0 for $n \geq R_1$. When the buffer occupancy is decreasing and crosses the congestion discard threshold R_1, the signalling traffic load has value λ', and after the buffer occupancy crosses the congestion abatement threshold L_1, the signalling traffic load has value λ.

In the next section we show how the ideas of hysteretic control implemented in SS7 are applicable to solution of overload control problem in SIP server networks.

3 THE PROBLEM OF CONGESTION CONTROL IN THE NETWORK OF SIP SERVERS

With the increasing number of users of services based on SIP protocol, different types of SIP-server overloads arise due to lack of resources for user agent registration and for session establishment and termination.

There are two types of overload: a client to server overload ("client-to-server") or a server to server overload ("server-to-server") [5]. "Client-to-server" overload appears in SIP server (Registration server) due to excessive load created by large groups of SIP terminals. This overload can happen when power is restored after a mass power failure in a large metropolitan area, and after the power is restored, a very large number of SIP devices boot up and send out SIP registration requests almost simultaneously, which could easily overload the corresponding SIP registration server. "Server-to-server" overload appears in the SIP server (Proxy server) as a result of some special events, also referred to as flash crowds.

The SIP protocol provides a basic overload control mechanism through the 503 (Service Unavailable) response code [3]. SIP servers that are unable to forward a request due to temporary overload can reject the request with a 503 response. The overloaded server can insert a Retry-After header into the 503 response, which defines the number of seconds during which this server is not available for receiving any further request from the upstream neighbour. A server that receives a 503 response from a downstream neighbour stops forwarding requests to this neighbour for the specified amount of time and starts again after this time is over. Without a Retry-After header, a 503 response only affects the current request and all other requests can still be forwarded to this downstream neighbour. A server that has received a 503 response can try to re-send the request to an alternate server, if one is available. A server does not forward 503 responses toward the UA and converts them to 500 Server Internal Error responses instead.

The problems that arise as a result of overload control mechanism of the SIP server using the 503 (Service Unavailable) message are listed in [5]. Note that in modern IETF standards these problems are still unsolved.

Depending on the method by which the sender determines the state of the receiver and manages the load, the congestion control mechanisms can be divided into explicit overload control mechanisms that are in fact feedback based, and implicit overload control mechanisms that are self-limiting.

There are three explicit overload control feedback schemes formulated in [5]: Window-based Overload Control, Signal-based Overload Control, On-/Off Overload Control. The first two of them are now under intensive study in IETF [6, 7]. In our paper we focus on LBOC scheme. According to LBOC the receiver should request the sender to reduce the load on a given number of percentages, which is calculated by the recipient taking into account its current load.

Let us briefly describe the main features of the SIP mechanism, which is needed to convey overload feedback from the receiving to the sending SIP server. Three different alternative feedback mechanisms – local, hop-by-hop, and end-to-end – are determined in RFC 6357 [5].

If overload control is implemented locally, the SIP server measures the current utility of its processor and makes a decision to select the messages that will be affected and determines whether they are rejected or redirected. In case of end-to-end overload control, all the receiving servers along the path of a request should measure the current utility of their processors and notify the sender of a request concerning overloading. All the receiving servers have to cooperate to jointly determine the overall feedback for this path. Each sending server implements the algorithms needed to limit the amount of traffic forwarded to the receiving server. Note that in [5], the local mechanism was recognized as ineffective and end-to-end mechanism was deemed difficult to implement.

Hop-by-hop overload control does not require that all SIP entities in a network support it. It can be used effectively between two adjacent SIP servers if both servers support overload control and does not depend on the support from any other server or user agent. The more SIP servers in a network support hop-by-hop overload control, the better protected the network is against occurrences of overload. Therefore, overload control is best performed hop-by-hop. The receiving SIP server monitors the current utility of its processor and notifies the sending server in case of overloading. The sending server acts on this feedback and reduces the outgoing load, for example, by rejecting messages if needed. According to the LBOC scheme [6], a server asks an upstream neighbour to reduce by the desired percentage the number of requests it would normally forward to this server. For example, a server can ask an upstream neighbour to reduce the number of requests by 10 %. The upstream neighbour then redirects or rejects the messages that are destined for this server with dropping probability $q = 0.1$. The alternative is a RBOC which is defined in [7]. When the rate-based overload control mechanism is used, a server notifies an upstream neighbour to send requests at a rate not greater than or equal to the desired number of requests per second.

The above hop-by-hop overload control principles have been used as the basis of a simulation model and for formulation of the optimization problem of hop-by-hop overload control. In addition, analytical formulae were developed to support the simulation. We considered the interaction between two adjacent SIP servers that use LBOC scheme and built a queueing model with the aim of analysing the control parameters using the hysteretic load control idea from [20].

Below we briefly discuss and introduce all necessary notations and main performance measures which are useful for overload control analysis.

We consider a queueing system where customers arrive and receive service in accordance with the overload control algorithm. The server operates in three modes: normal ($h = 0$), overload ($h = 1$), and discard ($h = 2$), where h is the overload status. When the queue length increases and exceeds the threshold, H, in the normal mode, the system detects the overload and switches to the overload mode. In the overload mode, the system reduces input flow: newly arriving customers are discarded with dropping probability, q. Thereafter, if the queue length decreases and drops below the threshold, L, in the overload mode, the system detects the elimination of overload, turns to normal mode and starts to put all newly arrived customers into the queue. If in the overload mode the queue length continues increasing and reaches threshold, R, the system turns to the discard mode and all newly arrived customers are discarded. After that, the queue length starts decreasing in the discard mode and when it drops below the threshold, H, the system detects mitigation of overloading, turns to the overload mode and starts to put newly arrived customers into the queue with probability $p = 1 - q$.

Let n denote the queue length, $n = 0, \ldots, R$. Then the state space of the system is of the form

$$\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1 \cup \mathcal{X}_2, \tag{1}$$

where $\mathcal{X}_0 = \{(h,n) : h = 0, 0 \le n \le H-1\}$ is the set of states of normal load, $\mathcal{X}_1 = \{(h,n) : h = 1, L \le n \le R-1\}$ is the set of overload states, and $\mathcal{X}_2 = \{(h,n) : h = 2, H+1 \le n \le R\}$ is the set of discard states.

Then the input load function $\lambda(h,n)$ is shown in Figure 3 and specified by the following relation:

$$\lambda(h,n) = \begin{cases} \lambda, & (h,n) \in \mathcal{X}_0, \\ p\lambda, & (h,n) \in \mathcal{X}_1, \\ 0, & (h,n) \in \mathcal{X}_2. \end{cases}$$

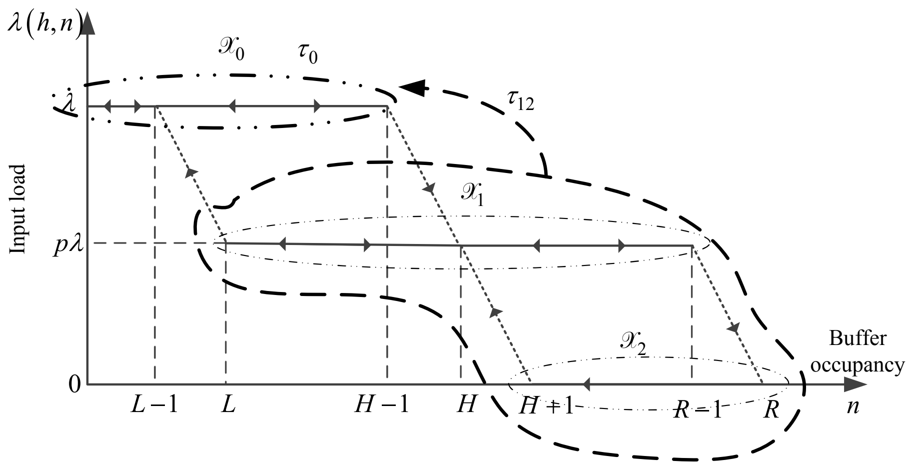


Figure 3. Function of the input load $\lambda(h,n)$

Figure 3 schematically illustrates set $\mathcal{X}$ and an important performance measure – random variable τ_{12} of the return time from the set of overload states $\mathcal{X}_{12} = \mathcal{X}_1 \cup \mathcal{X}_2$ in the set of the normal load states $\mathcal{X}_0$. Let τ_0 denote the time duration of the system in set $\mathcal{X}_0$. Then variable $\tau = \tau_0 + \tau_{12}$ is the control cycle time. For further analysis, we also need the following variables: the probability of the system being in the set of normal load states is denoted by $P(\mathcal{X}_0)$, the probability of being in the set of overload states is denoted by $P(\mathcal{X}_1)$, and the probability of being in the set of discard states is denoted by $P(\mathcal{X}_2)$.

Next, we analyse the queueing model of the LBOC scheme, and derive formulas for the analysis of its key performance measures.

4 LOSS-BASED OVERLOAD CONTROL PERFORMANCE ANALYSIS

There are a number of papers [8, 9, 10, 11, 12, 13, 15] which describe approaches to building models with a threshold overload control; but neither in the IETF standards nor in other available sources there are any analytical models for SIP-server overload control. So in this section we construct a model of SIP server with LBOC scheme on the principles of the SS7 hysteretic load control introduced in Section 1. Here we investigate one of four generic SIP server configurations [5], the so-called "multiple sources" configuration, in which the downstream server receives traffic from K upstream servers. Each of these servers can contribute by a different amount of traffic, which can vary over time.

Figure 4 shows the system model of LBOC implemented between each "upstream–downstream" server pair. The model identifies components of LBOC that is proposed to be implemented using a hysteretic technique. We develop the model in accordance with the RFC 6357 recommendations on design considerations for explicit SIP overload control, and include the following components: SIP Processor, Monitor, Control Function, and Actuator. The Processor is protected by an overload control mechanism and includes a buffer where SIP messages are queued according to the FIFO order. The Monitor measures the Processor load in the Receiving Entity (RE) and reports the buffer occupancy $n = (n_1, \ldots, n_K)$ to the Control Function, where n_k is the number of messages in the input buffer received from the k^{th} Sending Entity (SE). The Control Function uses the buffer occupancy to determine whether an overload is likely to occur according to the hysteretic load control algorithm, and identifies the limitation required to adjust the load sent to the Processor on the RE. Thus, the Control Function in the RE sends the dropping probability q to the SE.

Note that the Control Function in the RE realizes two algorithms. The first one determines whether an overload will occur, and the second one calculates the dropping probability $0 < q_k \leq 1$ for throttling of traffic at the SE. According to RFC 6357, the Control Function at the SE is empty and simply passes q_k along as feedback to the Actuator. The Actuator implements the throttling algorithm for traffic forwarded to the RE.

Further, under the assumption of a stationary Poisson input, we construct a model of the LBOC scheme in a SIP server using a hysteretic technique. Without loss of generality of the model, we assume that $K = 2$. The model is represented by a single-server queuing system with a finite buffer of capacity R, as shown in Figure 5. Note that the case of $R = \infty$ was studied in [14]. Two Poisson flows

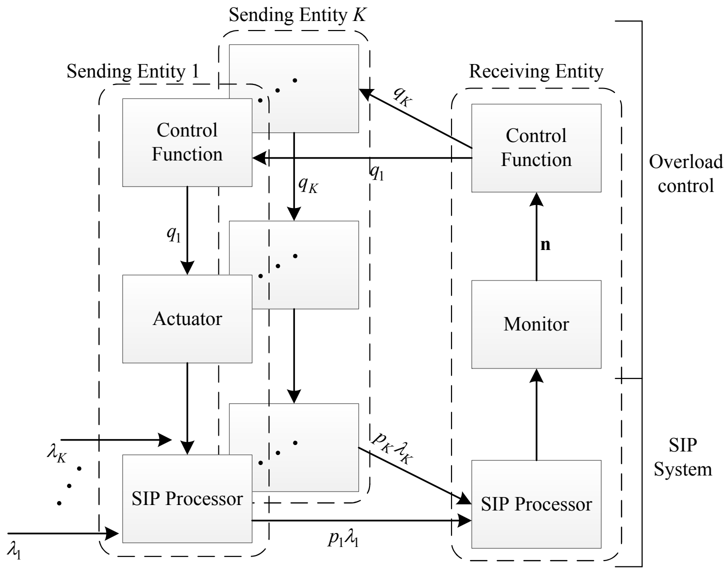


Figure 4. System model for the LBOC scheme

reach the system with an intensity of $\lambda_k(h,n)$. Under normal load conditions, the intensity of the k^{th} incoming flow is equal to $\lambda_k > 0$ and, if the system is in an overload condition, the flow rate decreases such that $\lambda'_k = p_k\lambda_k$, $0 < p_k < 1$. If the system is in a discard state, the intensity of the incoming flow is equal to zero.

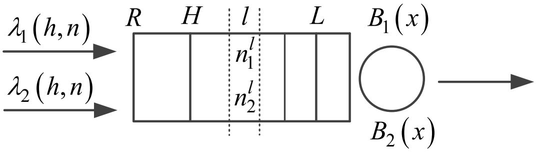


Figure 5. Queuing model of $M_2|G_2|1|\langle L,H\rangle|R$ with hysteretic load control

Both customer types are served by a single server on a FCFS basis, and let $B_k(t)$ be the generic service-time distribution function of the k^{th} flow. Let $\left(n_1^l, n_2^l\right) \in \{(0,0),(0,1),(1,0)\}$ indicate the state of the l^{th} position of the buffer, i.e., $n_i^l = 1$ if the l^{th} position is occupied by the k^{th} customer, and $n_k^l = 0$ otherwise. The total number of customers of both types can then be calculated by the formula $n = n_1 + n_2 = \sum_{l=1}^{R} n_1^l + \sum_{l=1}^{R} n_2^l$, where n_k is the number of the k^{th} customer in the buffer.

We assume that the customer being served retains a place in the queue, which is released when the service is terminated. We implement hysteretic load control of the input flows through two thresholds: an abatement threshold L, and an onset threshold H.

In the case of $B_1(t) = B_2(t) = 1 - e^{-\mu t}$, the queuing system can be described by Markov process $X(t)$ over the state space $\mathcal{X}$ defined by formula (1), and the transition diagram of the system can be pictured as in Figure 6, where $\lambda = \lambda_1 + \lambda_2$ and $\lambda' = \lambda'_1 + \lambda'_2$.

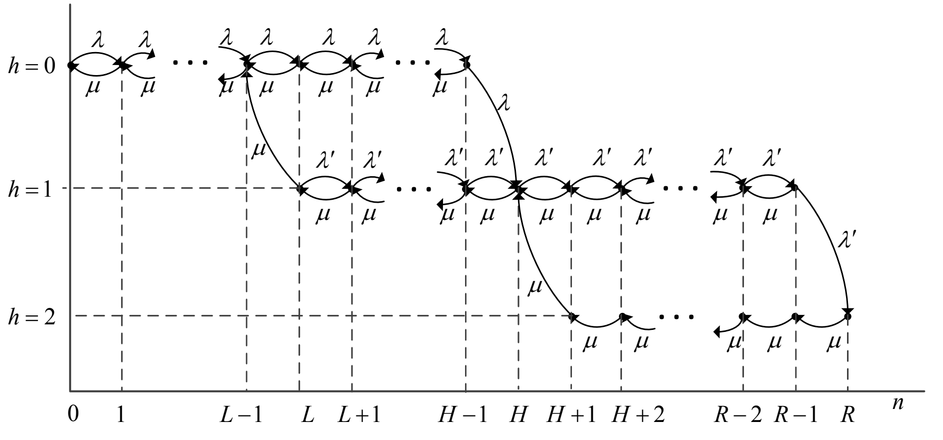


Figure 6. Transition diagram of Markov process $X(t)$

Let τ_{12} measure the time from the moment when the process $X(t)$ reached the overload set $\mathcal{X}_{12} = \mathcal{X}_1 \bigcup \mathcal{X}_2$, i.e., the state $(1, H+1)$, for the first time until it reaches the normal load set $\mathcal{X}_0$, namely the state $(0, L-1)$. Clearly, random variable τ_{12} is the return time to the normal load set. Below, we propose a method for calculating the probability distribution function of the random variable τ_{12} for an $M_2|M|1|\langle L, H\rangle|R$ system.

Let $\widehat{X}(t)$ be the truncation of process $X(t)$ to the set $\widehat{\mathcal{X}} = \mathcal{X}_1 \bigcup \mathcal{X}_2 \bigcup \{(0, L-1)\}$, and let $\widehat{p}_{hn}(t) = P\left\{\widehat{X}(t) = (h,n)\right\}$, $(h,n) \in \widehat{\mathcal{X}}$, $\widehat{\mathbf{p}}(t) = (\widehat{p}_{hn}(t))_{(h,n)\in\widehat{\mathcal{X}}}$, and $\widehat{\mathbf{P}}(t) = (\widehat{p}_{hn,rm}(t))_{(h,n),(r,m)\in\widehat{\mathcal{X}}}$, where $\widehat{\mathbf{p}}(t)$ is the row vector of state probabilities of Markov process $\widehat{X}(t)$ at the moment $t \geq 0$ and $\widehat{\mathbf{P}}(t)$ is the transition probability matrix in the interval $[0, t)$. The state transition diagram for the process $\widehat{X}(t)$ is shown in Figure 7.

It is known that matrix $\widehat{\mathbf{P}}(t)$ can be written in the form: $\widehat{\mathbf{P}}(t) = e^{\widehat{\mathbf{\Lambda}}t}$, where $\widehat{\mathbf{\Lambda}}$ is the infinitesimal operator of process $\widehat{X}(t)$. The vector $\widehat{\mathbf{p}}(t)$ for Markov process $\widehat{X}(t)$ then satisfies the following equation:

$$\widehat{\mathbf{p}}^T(t) = \widehat{\mathbf{p}}(0)\,\widehat{\mathbf{P}}(t) = \widehat{\mathbf{p}}(0)\,e^{\widehat{\mathbf{\Lambda}}t}. \tag{2}$$

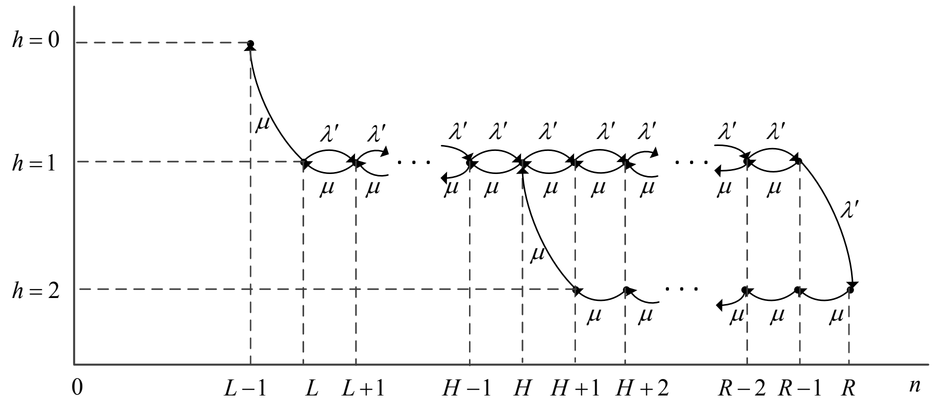


Figure 7. State transition diagram for truncated process $\widehat{X}(t)$

As part of the problem being solved, the initial probability vector $\widehat{\mathbf{p}}(0)$ can be written as:

$$\widehat{p}_{hn}(0) = \begin{cases} 1, \ (h,n) = (1,H), \\ 0, \ \text{otherwise}, \end{cases} \tag{3}$$

and then, the distribution function $F_\tau(t)$ of the random variable τ is given by:

$$F_\tau(t) = \widehat{p}_{0,L-1}(t). \tag{4}$$

Thus, the problem of finding the characteristics of the random variable τ (return time to the set of normal load states) is reduced to the calculation of the distribution function $F_\tau(t)$ using Equations (2) and (4), while the initial probability vector $\widehat{\mathbf{p}}(0)$ is given by Equation (3).

Thus, the problem of finding the characteristics of the random variable τ_{12} of the return time to the set of normal load states is reduced to the calculation of the distribution function $F(t)$ by Equations (2) and (4), while the initial probability vector $\widehat{p}(0)$ is given by Equation (3).

Note that calculation of mean return time $E\tau_{12}$ using probability distribution function is associated with certain computational difficulties and besides it gives suitable results only for system with exponential service times (i.e. for $M_2|M|1|\langle L,H\rangle|B$ system). To avoid these difficulties two fast algorithms for the computation of this important performance characteristic were developed: one of them is described in details in [19] and the other is given below. Let us introduce the following notation:

- m_n, $n = L, \ldots, B-1$ is the mean time to the moment when the number of customers in the system hits $L-1$ for the first time, given that at some moment there were n customers in the system, which accepted newly arriving customers with probability p;
- m_n^*, $n = H+1, \ldots, B$ is the mean time to the moment when the number of customers in the system hits $L-1$ for the first time, given that at some

moment there were n customers in the system, which discarded all newly arriving customers.

One can derive from [21] that m_n and m_n^* are the minimal nonnegative solution of the following system of linear algebraic equations:

$$
\begin{aligned}
m_L &= \frac{1}{\mu+\lambda_1} + \frac{\lambda_1}{\mu+\lambda_1} m_{L+1}, && (5)\\
m_n &= \frac{1}{\mu+\lambda_1} + \frac{\mu}{\mu+\lambda_1} m_{n-1} + \frac{\lambda_1}{\mu+\lambda_1} m_{n+1}, \quad n = L+1,\dots,R-2, && (6)\\
m_{R-1} &= \frac{1}{\mu+\lambda_1} + \frac{\mu}{\mu+\lambda_1} m_{R-2} + \frac{l_1}{\mu+\lambda_1} m_R^*, && (7)\\
m_n^* &= m_H + \frac{n-H}{\mu}, \quad n = H+1,\dots,R. && (8)
\end{aligned}
$$

If we introduce the following auxiliary variables

$$a = \frac{1}{\mu+\lambda_1}, \quad b = \frac{\mu}{\mu+\lambda_1}, \quad c = \frac{\lambda_1}{\mu+\lambda_1},$$

$$x_1 = \frac{1}{\mu+\lambda_1} + \frac{\lambda_1}{\mu+\lambda_1} \cdot \frac{R-H}{\mu}, \quad y_1 = \frac{\mu}{\mu+\lambda_1}, \quad z_1 = \frac{\lambda_1}{\mu+\lambda_1},$$

$$u_1 = \frac{a + c x_{D-H-1}}{1 - c y_{D-H-1} - c z_{D-H-1}}, \quad v_1 = \frac{b}{1 - c y_{D-H-1} - c z_{D-H-1}},$$

$$x_n = \frac{a + c x_{n-1}}{1 - c y_{n-1}}, \quad y_n = \frac{b}{1 - c y_{n-1}}, \quad z_n = \frac{c z_{n-1}}{1 - c y_{n-1}},$$

$$u_n = \frac{a + c u_{n-1}}{1 - c v_{n-1}}, \quad v_n = \frac{b}{1 - c v_{n-1}}, \quad \forall n \ge 2,$$

then the solution of (5) after tedious calculations can be represented as follows

$$
\begin{aligned}
m_L &= \frac{a + c u_{H-L}}{1 - c v_{H-L}}, && (9)\\
m_n &= u_{H+1-n} + v_{H+1-n} m_{n-1}, \quad n = L+1,\dots,H, && (10)\\
m_n &= x_{D-n} + y_{D-n} m_{n-1} + z_{R-n} m_H, \quad n = H+1,\dots,R-1, && (11)\\
m_n^* &= m_H + \frac{n-H}{\mu}, \quad n = H+1,\dots,R. && (12)
\end{aligned}
$$

Note the solution (9) consists of $B-L$ values and each of them is the mean time that takes the system to get back to the set of normal load states provided that it is in overload set and it is known how many customers are there in the system. The question is which one of these values is the mean return time $E\tau_{12}$ defined above. The answer depends on the moment of time when we start to observe the system. If we start to observe the system when it is in a state of normal load then $E\tau_{12} = m_H$;

but if we start to observe the system when its state belongs to overload set then $E\tau_{12}$ equals m_n or m_n^* depending on the state n of the system at the moment of time when the observation took place.

Thus we have all necessary expressions to carry out a case study. Its goal is to illustrate the control mechanism that minimizes the mean return time. We give a numerical example for the mean return time of $M|M|1$ and $M|D|1$ queues. We would like to note that for system with deterministic service times relatively fast computational algorithm for mean return time was also developed, but we do not give it here, limiting ourselves below to numerical results only.

In order to estimate the mean service rate μ of SIP messages, we have taken into account that each session involves the exchange of seven SIP messages. We assume that the processing time of an INVITE message is 10 ms and the processing time of a non-INVITE message is 5 ms. Taking into account that a basic session is composed of one INVITE and six non-INVITE messages, the average processing time of a SIP message is about 5 ms, hence $\mu = 200\,s^{-1}$.

The problem is stated as follows [17]: minimize the mean return time with respect to the choice of the two thresholds, L and H, such that requirements $R1-R3$ are satisfied. Formally:

$$\begin{aligned}
&E\tau_{12}\,(L,H) \to \min;\\
&R1:\ P\,(\mathcal{X}_1) \le \gamma_1;\\
&R2: P\,(\mathcal{X}_2) \le \gamma_2;\\
&R3: \tau \ge \gamma_3.
\end{aligned}$$

For a given dropping probability $q \in \{0.3, 0.4, 0.5, 0.6\}$, we now seek to solve the problem of choosing the two threshold values. Let us also consider minimizing the mean return time $E\tau_{12}$ such that the offered load $\rho = \lambda/\mu = 1.2$, buffer capacity $R = 100$, $\gamma_1 = 0.2$, $\gamma_2 = 10^{-4}$, and $\gamma_3 = 450$ ms. Using the above formulae, we developed an algorithm for solving the optimization problem. Note that for the optimum solution obtained by this algorithm, requirements $R1$ and $R2$ are always binding, making the mean control cycle time as high as possible. The results of calculations with the above-defined input data for exponential (M) and deterministic (D) service times are presented in Figure 8.

The graph shows that the mean return time for the $M|M|1$ system is about twice that for the $M|D|1$ system. Taking into account that six out of seven messages have almost the same length, we recommend the use of the $M|D|1$ system for analysis, i.e., to choose the thresholds according to the dashed curve.

5 OSN ARCHITECTURE DESIGN

Now we present the capabilities of our Open SIP signalling Node (OSN) which is under development and will support *nix like OS. Node software architecture can be logically divided into four levels as shown in Figure 9: OS/Network Level, Traffic Load Control Level, SIP System Level, Application Level.

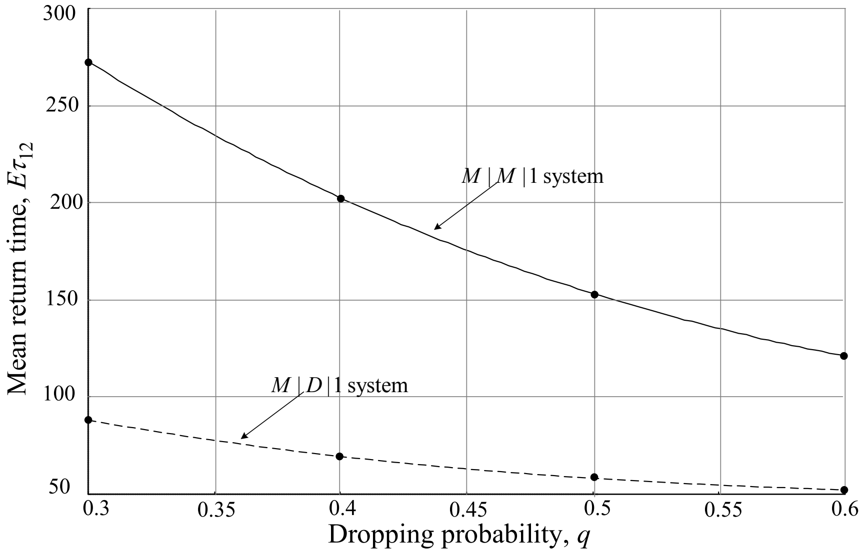


Figure 8. Mean return time for the $M|M|1$ and $M|D|1$ queues. Values of L and H are also shown.

OS/Network Level corresponds to the first four levels of the ISO/OSI model and is implemented in OS, e.g. CentOS.

Traffic Load Control Level consists of three modules: Actuator, Monitor and Control Function as shown in Figure 10.

Each module implements the functionality according to RFC 6357: "The Monitor measures the current load of the SIP Processor on the receiving entity. It implements the mechanisms needed to determine the current usage of resources relevant for the SIP Processor and reports load samples to the Control Function. The Control Function implements the overload control algorithm. The Control Function uses the load samples and determines if overload has occurred and a throttle needs to be set to adjust the load sent to the SIP Processor on the receiving entity. The Control Function on the receiving entity sends load feedback to the sending entity. The Actuator implements the algorithms needed to act on the throttles and ensures that the amount of traffic forwarded to the receiving entity meets the criteria of the throttle. The Actuator implements the algorithms to achieve this objective, e.g., using message gapping. It also implements algorithms to select the messages that will be affected and determine whether they are rejected or redirected."

The SIP System Level implements SIP baseline specifications RFC 3261 and RFC 3263. Support for other specification will be included later. The level includes SIP Manager, High and Low Message Processing modules, Presence Server and Proxy Core. New modules will be added if necessary. High and Low Message Processing modules provide the complete set of tools for processing SIP messages.

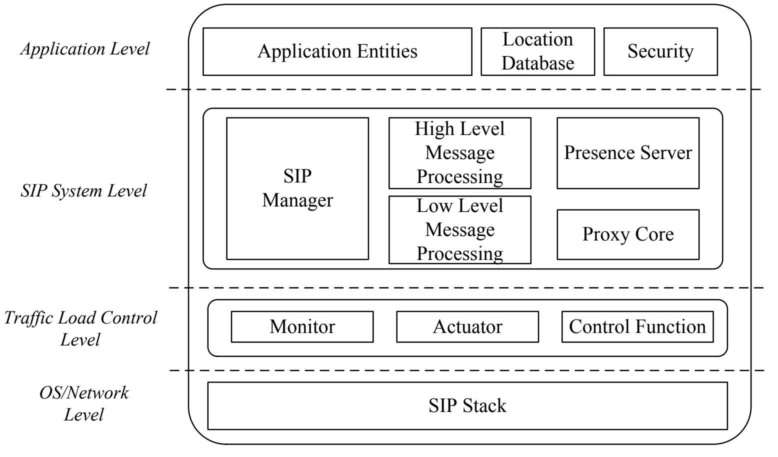


Figure 9. Open SIP signalling Node software architecture

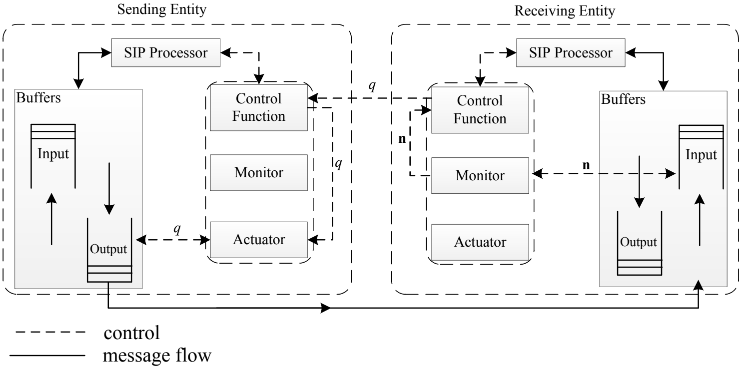


Figure 10. Loss-Based Overload Control scheme

The Presence Server module allows a party to know the ability and willingness of other parties to participate in a call even before an attempt has been made. The Presence Server is responsible for handling Presence SUBSCRIBE requests with event package “presence” from Watchers, and enables the application to notify them about the Presence status of the Presences. Proxy Core module implements the stateful and stateless proxy behaviour according to RFC 3261.

The Application Level implements all other aspects of the application, e.g., service engine, billing module and database access.

The OSN complies with the IETF specifications: RFC 3261, RFC 3263, RFC 6357, draft-ietf-soc-overload-control-08. Support for other specification may be included later. The OSN platform supports both basic (A records) and advanced

(SRV and NAPTR records) DNS queries as defined in RFC 3263 (Locating SIP Servers). OSN supports the full procedure for address resolution including determining a target-set (both Predefined and Defined by proxy based on location service or any other means) and DNS resolution.

OSN proxy features complete SIP proxy functionality with the following capabilities: stateful forwarding, stateless forwarding, forking – parallel/sequential/mixed, record-routing, loose-routing, CANCEL processing and forwarding, recursion on 3xx responses, loop detection, max forwards check, working as outbound proxy, message validation, authentication.

The OSN software supports all standard registrar functionalities: accept and validate REGISTER messages, read location mappings from location service, apply registration logic, update location service, remove location mappings that have expired from the location service, authentication.

The platform provides full redirect server functionality: receiving and processing incoming responses, address resolution, returning 3xx response with one or more contact addresses, authentication.

The OSN software implementation includes some non-SIP functionalities in the form of Server Components: Location service (database) and User/password database. The OSN Platform comes complete with default implementations for server components. These are provided as reference implementations only. Currently we are in need of the following server components:

1. Location database implements the interface to the location service (the storage place of SIP location mappings). The SIP server uses this interface to read and to write location mappings as part of the address resolution process (proxy and redirect servers) and the registration process (registrar). Implementations may vary according to the type of location database used. Possible implementations include LDAP client, SQL client.
2. Security component implements all non-SIP aspects of security, such as cryptographic algorithms and user/password databases. The default implementation provided with OSN implements MD5-hash and MySQL user/password database.

6 CONCLUSION

In this paper hysteretic load control technique is proposed to solve the problem of LBOC scheme implementation in SIP server software. The performance of the system depends on the dropping probability, onset threshold, abatement threshold, and discard threshold.

First, we model the system analytically as a queue with several input flows, that are being throttled depending on hysteretic load control.

Second, using the experience gained from the analytical modelling and numerical experiments, we design the OSN software architecture for simulation of LBOC scheme.

Future work will be focused on the software implementation, simulations of LBOC and RBOC schemes, and on further SIP overload control mechanism analytical modelling.

Acknowledgement

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

REFERENCES

[1] ITU-T Recommendation Q.704: Signalling System No. 7 – Message Transfer Part, Signalling network functions and messages, 1996.

[2] ITU-T Recommendation Q.752: Monitoring and measurements for Signalling System No. 7 networks, 1997.

[3] Rosenberg, J.—Schulzrinne, H.—Camarillo, H. et al.: SIP: Session Initiation Protocol. RFC 3261, 2002.

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

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

[6] Gurbani, V.—Hilt, V.—Schulzrinne, H.: Session Initiation Protocol (SIP) Overload Control. Draft-ietf-soc-overload-control-08, 2012.

[7] Noel, E.—Williams, P. M.: Session Initiation Protocol (SIP) Rate Control. Draftietf-soc-overload-rate-control-02, 2012.

[8] Ohta, M.: Overload Control in a SIP Signalling Network. International Journal of Electrical and Electronics Engineering, 2009, pp. 87–92.

[9] Hilt, V.—Widjaja, I.: Controlling Overload in Networks of SIP Servers. IEEE International Conference on Network Protocols, 2008, pp. 83–93.

[10] Shen, C.—Schulzrinne, H.—Nahum, E.: Session Initiation Protocol (SIP) Server Overload Control: Design and Evaluation. Lecture Notes in Computer Science, Vol. 5310, Springer, 2008, pp. 149–173.

[11] Garroppo, R. G.—Giordano, S.—Spagna, S.—Niccolini, S.: Queueing Strategies for Local Overload Control in SIP Server. IEEE Global Telecommunications Conference 2009, pp. 1–6.

[12] Montagna, S.—Pignolo, M.: Load Control Techniques in SIP Signalling Servers Using Multiple Thresholds. Proc. of 13th International Telecommunications Network Strategy and Planning Symposium 2008, pp. 1–17.

[13] Garroppo, R. G.—Giordano, S.—Niccolini, S.—Spagna, S.: A Prediction-Based Overload Control Algorithm for SIP Servers. IEEE Transactions on Network and Service Management, Vol. 8, 2011, No. 1, pp. 39–51.

[14] Roughan, M.—Pearce, C.: A Martingale Analysis of Hysteretic Overload Control. Advances in Performance Analysis: A Journal of Teletraffic Theory and Performance Analysis of Communication Systems and Networks, Vol. 3, 2000, pp. 1–30.

[15] Takagi, H.: Analysis of a Finite-Capacity M/G/1 Queue with a Resume Level. Performance Evaluation, Vol. 5, 1985, pp. 197–203.

[16] Gebhart, R. F.: A Queuing Process with Bilevel Hysteretic Service-Rate Control. Naval Research Logistics Quarterly, VINITI, Vol. 14, 1967, pp. 55–68.

[17] Yum, T. P.—Yen, H. M.: Design Algorithm for a Hysteresis Buffer Congestion Control Strategy. IEEE International Conference on Communication, Boston, Massachusetts 1983, pp. 499–503.

[18] Abaev, P.—Gaidamaka, Y.—Samouylov, K.: Modeling of Hysteretic Signalling Load Control in Next Generation.: Proc. of the 12th International Conference on Next Generation Wired/Wireless Networking (Winter Session), Lecture Notes in Computer Science, Springer 2012.

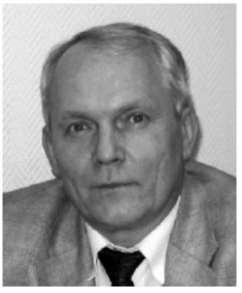
[19] Abaev, P.—Gaidamaka, Y.—Pechinkin, A.—Razumchik, R.—Shorgin, S.: Simulation of Overload Control in SIP Server Networks. Proceedings of the 26th European Conference on Modelling and Simulation (ECMS) 2012, pp. 533–539.

[20] Abaev, P.—Gaidamaka, Y.—Samouylov, K.: Analytical Model of Loss-Based Overload Control Algorithm in SIP Server Using Hysteretic. Proceedings of the 12th International Conference on Next Generation Wired/Wireless Networking (Summer Session), Lecture Notes in Computer Science, Springer 2012.

[21] Kelbert, M.—Suhov, Yu.: Probability and Statistics by Example. Volume II – Markov chains: A Primer in Random Processes and Their Applications. Independent University Press, Moscow 2010 (in Russian).

Pavel Abaev received his Ph. D. in Computer Science from the Peoples' Friendship University of Russia in 2011. From 2011 to 2013, he was a Senior Lecturer in the university. Since 2013 he has been an Associate Professor at Telecommunication Systems Department of the university. His current research focuses on NGN signalling, QoS analysis of SIP, and mathematical modeling of communication networks.

Yuliya 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 Telecommunication Systems Department of the university. 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.



Konstantin Samouylov received his Ph. D. from the Moscow State University and his 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 the Head of the Telecommunication Systems Department in 1996. His current research interests include 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.

Alexander 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.

Rostislav 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.

Sergey Shorgin received his Doctor of Sciences degree in physics and mathematics in 1997. Since 1999, he is the Deputy Director of the Institute of Informatics Problems, Russian Academy of Sciences; since 2003 he is a Professor. He is the author of more than 100 scientific and conference papers and a co-author of three monographs. His research interests include probability theory, modeling complex systems, actuarial and financial mathematics.