Overload Control in a SIP Signaling Network

Masataka Ohta

Abstract**—** The Internet telephony employs a new type of Internet communication on which a mutual communication is realized by establishing sessions. Session Initiation Protocol (SIP) is used to establish sessions between end-users. For unreliable transmission (UDP), SIP message should be retransmitted when it is lost. The retransmissions increase a load of the SIP signaling network, and sometimes lead to performance degradation when a network is overloaded.

The paper proposes an overload control for a SIP signaling network to protect from a performance degradation. Introducing two thresholds in a queue of a SIP proxy server, the SIP proxy server detects a congestion. Once congestion is detected, a SIP signaling network restricts to make new calls. The proposed overload control is evaluated using the network simulator (ns-2). With simulation results, the paper shows the proposed overload control works well.

Keywords**—** SIP signaling congestion overload control retransmission throughput simulation

I. Introduction

THE Internet telephony is experiencing significant growth providing low-price long distance calls. The Internet telephony employs a new type of Internet communication on which a mutual real-time communication is realized by establishing sessions between end-users. To establish the sessions, Session Initiation Protocol (SIP)[1] has been standardized by the Internet Engineering Task Force (IETF) as RFC3261[2]. Current applications of SIP focus on interactive multimedia sessions such as Internet telephony and multimedia conferences, but SIP or extensions of the protocol can also be used for instant messaging, event notification or managing other session types. It is expected that the number of new Internet services which employ the SIP will grows in the future.

For reliable transmissions, SIP messages should be transmitted over TCP. However, SIP messages sometimes have to be transmitted over UDP owing to capability of end devices. UDP is unreliable in nature. In order to keep high reliable transmissions of SIP messages in the Internet, SIP requests are retransmitted when adequate responses are not received in a predetermined interval. Although the retransmission is useful for maintaining the reliability, the retransmission increases load and can causes performance degradation of a SIP signaling network[3], [4].

This paper proposes an overload control to protect a SIP signaling network from a performance degradation when an overload is applied. An overload control is implemented in each protocol layer, namely the data link, network, transport and application layer. The SIP is a protocol in the application layer. So, the paper considers an overload control in the application layer (namely SIP layer).

The paper evaluates performance of the proposed overload control using the network simulator (ns-2). In order to simulate a SIP based signaling network, we have developed new types of agent which act as User agents and SIP proxy servers in the ns-2.

Prior works are as follows. The performance of SIP based network has been studied. [5] has studied voice quality effects of packet loss, delay and delay variation in a voice over IP (VoIP). [6] has studied how multilevel communication services can be guaranteed for multiple VoIP class. These studies focus on the real-time transport protocol (RTP) traffic which carries voice signals through established sessions. This paper focuses on SIP signaling traffic rather than RTP traffic. [7] has studied SIP signaling traffic. [7] has evaluated three transport protocols, UDP, TCP and SCTP (Stream Control Transmission Protocol), and has shown which protocol suites for carrying SIP messages. [8] has evaluated a call setup delay which is a key and easily discernable QOS parameter. [4] focuses on the retransmission and has shown a SIP proxy server configuration to face to a retransmission storm. [9] has studied performance of SIP network elements, such as SIP proxy servers, and has evaluated internal processing structures. They have evaluated string handling, and memory allocation, and thread architecture of SIP proxy servers. [10] has considered an overload control in the SIP layer. However, the paper has not shown a performance. This paper proposes an overload control, and evaluates a performance of the proposed scheme.

II. SIP Signaling

A. Outline of SIP Signaling

SIP is an application-layer control protocol that can establish multimedia sessions. Figure 1 shows a typical configuration for the SIP. In advance of establishing a session between caller (user-A in Fig.1) and callee (user-B in Fig.1), user agents (caller and callee) exchange information required for establishing a session through SIP signaling. SIP signaling is performed by sending requests and responses via SIP proxy servers. The routes of requests and responses are independent from routes of the established sessions. The signaling of SIP is took place between the neighbors which is shown by ①,②,③ in Figure 1. Resolving the SIP URI, each SIP proxy server

Masataka Ohta is with Faculty of Business Administration, Kanagawa University, 2946 Tuchiya, Hiratuka, Kanagawa, Japan (phone: +81-463-59-4111(ext.2215), fax: +81-463-58-9688, e-mail:m-ohta@kanagawa-u.ac.jp).

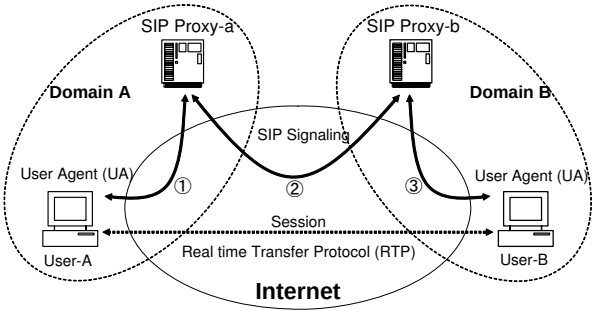

Fig. 1 Network configuration for SIP

performs routing of SIP requests and responses.

Figure 2 shows the typical SIP message exchange to establish a session. User-A calls user-B using user-B's SIP URI. As shown Figure 2, the request "INVITE" is used to request establishing a session between user-A and user-B. The node which receives an INVITE returns a provisional response 100Trying immediately indicating receipt of the INVITE and call progress. When user-B receives an INVITE, it checks and confirms parameters which need to establish a session. If user-B decides that the parameters are appropriate, it returns a response 180Ringing. When user-B answers, it sends a response 200OK. Finally user-A receives the 200OK and returns an ACK to user-B. Then, a session is established. Once a session has been established, both of user-A and user-B send media packets through the session. The request "BYE" is used to clear up the session.

B. Retransmission of SIP Messages

The SIP has two types of retransmission procedures, one for INVITE transaction shown with ① in Figure 2 and the

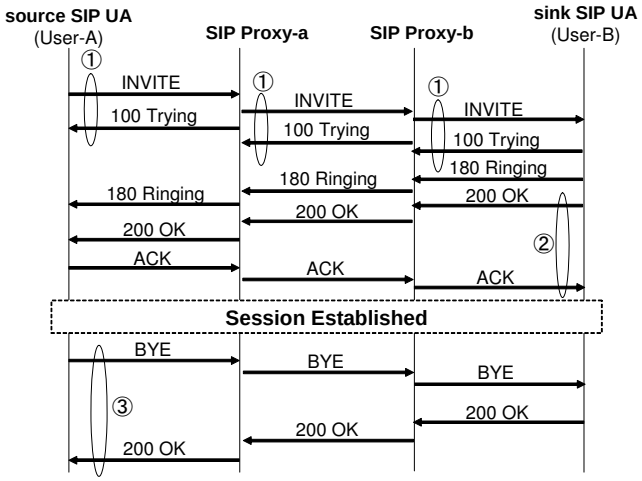

Fig. 2 A typical SIP message exchange

others for non-INVITE transactions (200OK, BYE). RFC 3261[2] defines the timer T1 for the retransmission. The client transaction retransmits an INVITE request at an interval that starts at T1 seconds, and the interval is doubled after each packet transmission. A client transaction ceases retransmission when it receives a provisional response, or when 64×T1 sec is passed after the initial transmission. Default value for T1 is 500ms[2]. So, after 32 sec in total the client transaction ceases retransmission when no response is received.

The retransmission procedure of non-INVITE transactions, namely 200OK shown with ② in Figure 2 and BYE shown with ③ in Figure 2, is somewhat different from that of INVITE transaction. RFC 3261[2] introduces another timer T2. Requests are retransmitted at T1 seconds, doubling the interval for each packet, and capping off at T2 seconds. This means that after the first packet is sent, the second is sent T1 seconds later, the next 2×T1 seconds after that, the next 4×T1 seconds after that, and so on, until the interval hits T2. Subsequent retransmissions are spaced by T2 seconds. Retransmission is ceased when 64×T1 sec is passed after the initial transmission, or when it receives a definitive response. Default value for T2 is 4 seconds[2]. After 32 seconds in total, the client transaction ceases if no response receives.

As shown here, the INVITE request is retransmitted up to 7 times in total, and 200OK and BYE requests are retransmitted up to 11 times in total. These retransmissions can degrade the performance of the SIP signaling. The paper shows how to protect these retransmission to improve a performance.

The paper uses throughput as a measure of performance. In the paper, the throughput is defined as rate of call completion that is number of calls which complete the entire message flow shown by Figure 2 in a second.

III. Simulation Model

A. Network Configuration

The paper considers the SIP signaling network shown in Figure 3. The source SIP user agents (UAs) shown by small squares of the left side of the figure are connected to routers shown by circles. SIP proxy servers shown by gray squares are also connected to the routers. Areas surrounded by the dotted circles show domains. As shown the figure, every domain $0 \sim n-1$ contains *m* source SIP UAs and one SIP proxy server. Domain *n* contains $n \times m$ sink SIP UAs and one SIP proxy server. Routers $0 \sim n-1$ are connected to router n. The $n \times m$ sink SIP UAs are shown by small squares of the right side of the figure. The routes of sessions are independent from that of the SIP messages. Dotted lines in Figure 3 indicate sessions. In the paper, we focus on the SIP signaling rather than media packets carried by sessions.

n×*m* pairs of source SIP UAs and sink SIP UAs are assumed to try establishing sessions. Since every sink UA belongs to

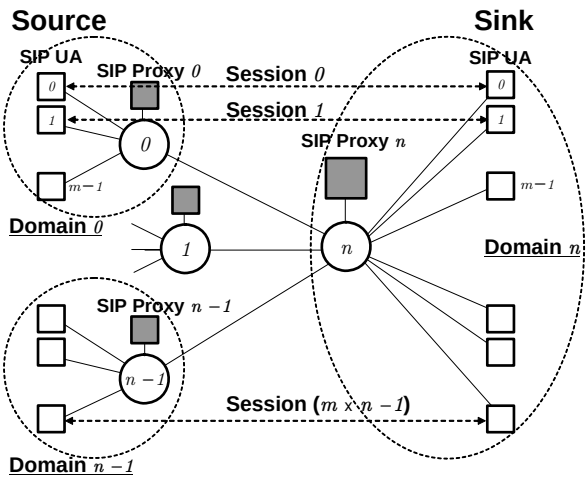

Fig. 3 Network model

the same domain (Domain *n* in Fig.3), every SIP messages is transferred to the SIP proxy server *n*. Consequently, SIP proxy server *n* is expected to be bottleneck in this situation.

In the study, speed of each link which connect UAs, SIP proxy servers and routers is assumed to 100Mbps. The paper also assumes that the average SIP message size is 731 bytes. Then the average processing time required for the SIP message transmission is 0.058 msec.

The paper evaluate the throughput of the SIP signaling in the network shown in Figure 3. Namely, the paper evaluates how many calls can be handled by the network in a second.

B. Overload Control

Figure 4 shows a queuing structure of the SIP proxy server. As shown the figure, the queue is a simple single queue. Every arrived SIP message is placed into the queue, and served with first-in first-out (FIFO) manner. The processing time of the SIP proxy server depends on the types of SIP messages. Usually, the processing time of the INVITE message is larger than that of other types of SIP messages, because of a query of a Data Base resolving a SIP URI. In the study, the processing time of the INVITE is assumed to 11.64 msec, and the average processing time is assumed to 2.6 msec (the processing times of each types of SIP messages are assumed for the study).

To detect an overload, we introduce two thresholds h and l. If the occupied number of buffer of the queue exceeds the threshold h, the SIP proxy server recognizes detecting a congestion. After that, if the occupied number of buffer becomes to be lower than l, the SIP proxy server recognizes that the congestion is removed. Figure 5 shows a state transition diagram. When a SIP message arrives at a SIP proxy server, it checks the occuipied number of buffer which is denoted by x in the figure. The congestion state transits according to the

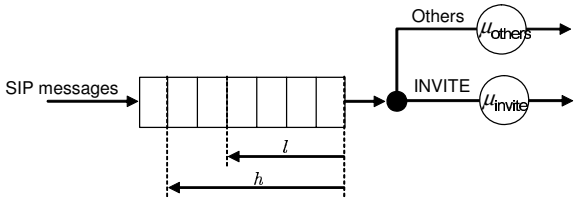

Fig. 4 Queuing structure and thresholds

current state and x.

When the SIP proxy server is in the congestion state, the SIP signaling network regulates to accept a new call. Figure 6 shows a message flow for the input regulation. Usually, a SIP proxy server returns the response " 100 " for "INVITE". As shown the figure, the SIP proxy server returns "503" (i.e., service unavailable) when the state is in the congestion. According to RFC 3261, when source SIP UA (client transaction) recievs "300-699" responce, it must stay a state starting Timer D which is defined in RFC 3261. The souce SIP UA can not send send any new INVITEs in this state. The period staying this state is controlled using Timer D . The value of Timer D is chosen 32 sec as default. Regulating making new calls by timer D, offered load to the network can be reduced. Then, it is expected that the overload is removed temporally.

C. Traffic Model

We assume that inter arrival time of call is T_I sec. Namely, a Source UA makes another call in T_I sec after the UA finishes a call. A sink UA is assumed to answer the call in T_A sec after the sink UA begins to ring. The duration of session is assumed to T_S sec. These values are exponentially distributed. So, every source UA is make a call every T_{call} sec where

$$T_{call} = T_I + T_A + T_S$$

As shown Figure 2, 7 SIP messages (INVITE, 100, 180, 200, ACK, BYE and 200) arrive to a SIP proxy sever ,and are served to complete a call. Since every SIP message is transffered to the SIP proxy n, average SIP message arrival

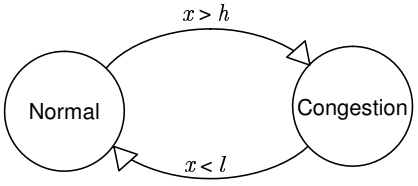

Fig. 5 State transition diagram for overload control

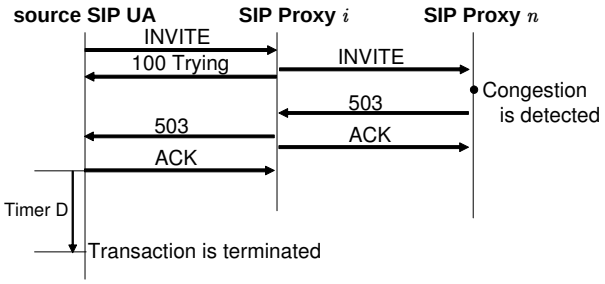

Fig. 6 Message flow for inpur reguration

rate to the SIP proxy sever n λ is

$$\begin{aligned}\lambda &= \frac{1}{T_{call}} \times 7 \times n \cdot m \\ &= \frac{7 \cdot n \cdot m}{T_I + T_A + T_S}\end{aligned}$$

where $n \cdot m$ is the total number of sources. We also introduce service rate of the SIP proxy server μ_{sip}. Namely, the SIP proxy server serves a SIP message in $1/\mu_{sip}$ sec which is the average processing time of SIP messages. Usage rate of the SIP proxy server n ρ_n can be expressed as

$$\rho_n = \frac{\lambda}{\mu_{sip}} \quad (1)$$

$$= \frac{7}{(T_I + T_A + T_S) \cdot \mu_{sip}} \cdot n \cdot m \quad (2)$$

In the study, we assume that $T_I = 30$ sec, $T_A = 4$ sec, $T_S = 30$ sec. So every source UA makes a call every 64 sec. $1/\mu_{sip}$ is also assumed to be 2.6×10^{-3} sec which means that capacity of the SIP proxy server is 195.7KBHC (Busy Hour Calls), and the maximum throughput of the SIP proxy server is 54.4 [calls/sec]. Based on the assumed parameters, the offered load to the SIP proxy server n is calculated as

$$\rho_n = 0.04 \times n \cdot m \times 10^{-3} \quad (3)$$

The offered load of the SIP proxy server $0 \sim n-1$ denoted by ρ_i is

$$\rho_i = 0.04 \times m \times 10^{-3} \qquad (i = 0 \sim n-1) \quad (4)$$

As explained in III-A, the average processing time of the link is assumed to 0.058 msec. So, the usage rate of the link which connects the SIP proxy n and the router n denoted by ρ_{link} is calculated as

$$\rho_{link} = \lambda \times 0.058 \times 10^{-3} \quad (5)$$

$$= \frac{7 \cdot n \cdot m}{T_I + T_A + T_S} \times 0.058 \times 10^{-3} \quad (6)$$

$$= 0.63 \times n \cdot m \times 10^{-5} \quad (7)$$

Since $\rho_n \geq \rho_i$ and $\rho_n > \rho_{link}$, we expect the SIP proxy server n is a bottleneck in this situation.

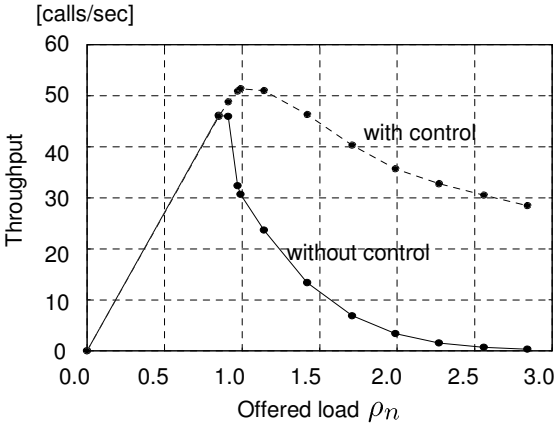

Fig. 7 Throughput characteristics

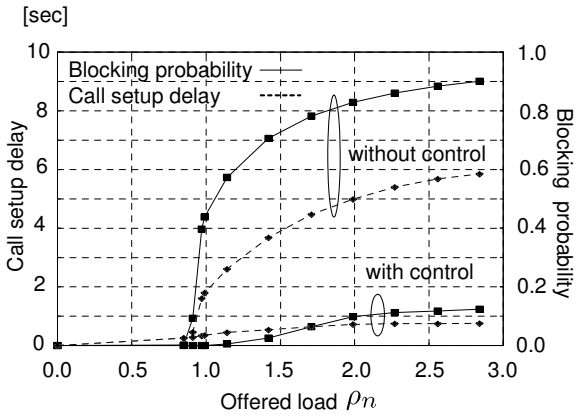

Fig. 8. Call setup delay and blocking probability

D. User Behavior

From a user's perspective, the call setup delay is important. The call setup delay is defined as the interval between entering the last dialed digit and receiving ring back in the telephony service. If the call setup delay is too long, user may abandon a call. In the study, user is assumed to abandon a call after the time T_{abdn} sec if user does not hear a ring back tone. T_{abdn} is assumed to be normally distributed. In the study, the average value and standerd deviation of T_{abdn} are assumed to 20 sec and 20/3 respectivelly.

Usually, users may retry to make a call after he abandons a call. However, the paper does not consider this retry to simplify the situation.

IV. Evaluation of Overload Control

Under the condition described in Section III, the paper evaluates the throughput of the network using the network simulator (ns-2) [11].

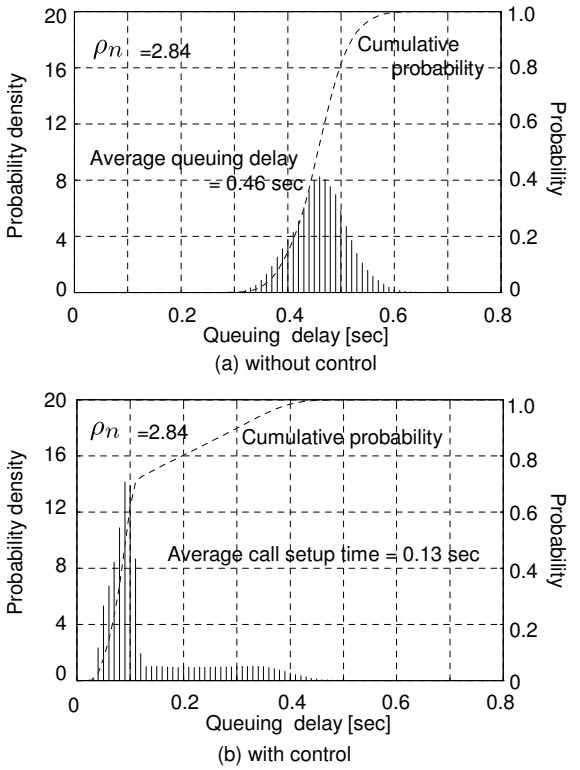

Fig. 9 Queuing delay distribution

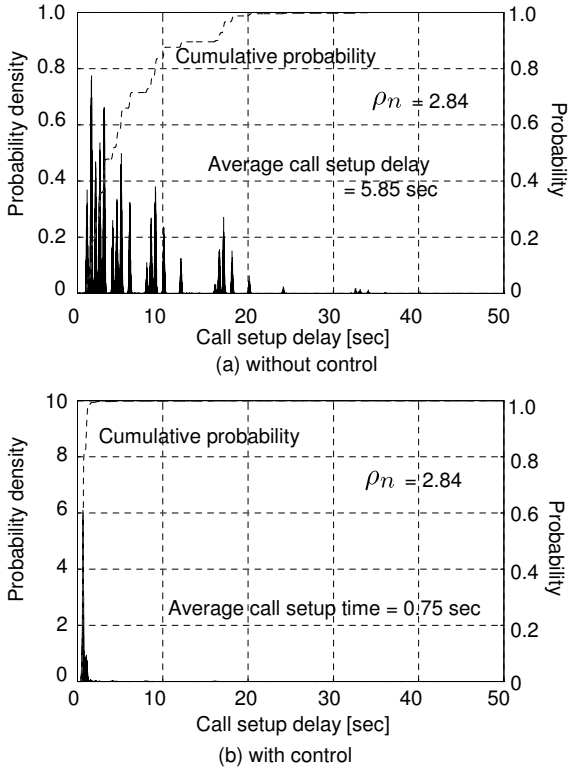

Fig. 10 Probability distribution of Call setup delay

In the simulation study, we have assumed the following parameters

- T1 = 0.5 sec, T2 = 4.0 sec
- Average call abandon time (T_{abdn}) = 20 sec
- No. of buffer = 100
- h = 80, l = 40

These parameters are common to all figures in this section.

Figure 7 shows throughput characteristics. The horizontal line is offered load ρ_n which is given by eq.(3). In the figure, we assume that T1 = 0.5 sec and T2 = 4 sec which are the default values in RFC3261. The total number of buffer of the queue is also assumed to 100, and the threshoulds h and l are assumed to 80 and 40 respectively. As shown the figure, throughput without overload control decreases when the offered load ρ_n exceeds 1.0. Namely, when the SIP signaling network is overloaded, the throughput decreases and becomes to be almost 0 for $\rho_n > 2.5$. The figure also shows the throughput characteristic when the overload control is applied. We can see that the throughput characteristic is improved significantly and the overload works well. The throughput for $\rho_n = 2.84$ is improved from 0.31 without control to 28.48 [call/sec] by the control.

Let us see what happen inside of the network. Figure 8 shows the call setup delay and blocking probability of buffer of the SIP proxy server n. The parameters are the same as in Figure 7. Both of call setup delay and blocking probability increase as the offered load ρ_n increases. The call setup delay and the blocking probability are 5.85 sec and 0.901 respectively in case of without control and ρ_n = 2.84. With the overload control, these values are improved significantly. The call setup delay and the blocking probability are 0.75 sec and 0.123 respectively when ρ_n = 2.84. For the sake of control, the call setup delay remains small even if an overload is applied.

Figure 9 shows probability distributions of queuing delay of the SIP proxy n. We can see that the overload control reduces the queuing delay. The average queuing delay is reduced from 0.46 sec without control to 0.13 sec with control. Figure 9(a) shows that the shape of the distribution is symetirical when the control is not applied. When the control is applied, the distribution has a long tail toward large values of the delay. But, the value does not exceed 0.5 sec.

Figure 10 shows probability distribution of the call setup delay. As shown Figure 10(a), the call setup delay widely varies and has a large delay variation if the control is not applied. The delay can be larger than 20 sec. On the other hand, Figure 10(b) shows that the distribution of the the call setup delay. In contrast to Figure 10(a), the delay variation is quite small. There is no probability that the delay is larger than 2 sec. The call setup delay strongly depends on the blocking

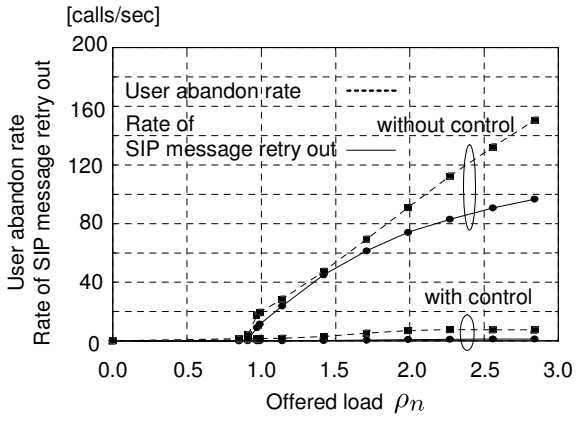

Fig. 11 Rate of unsuccessful call

probability of buffer. Let b be the blocking probability of buffer, and letting P_i be the plobability that "INVITE" is placed into the queue at i-th retransmission,

$$P_i = (1-b)b^{i-1} \tag{8}$$

The interval from the first transmission of "INVITE" to i-th transmission I_i can be written as

$$I_i = T1 \times (2^i - 1) \tag{9}$$

So, the mean interval from the first transmission to the sucessfun transmission D is

$$D = \sum_{i=1}^{7} I_i \cdot P_i \tag{10}$$

Using eq.(8) and eq.(9), D is

$$D = T1 \cdot (1-b)\{2\frac{1-(2b)^7}{1-2b} - \frac{1-b^7}{1-b}\} \tag{11}$$

Roughlly speaking,

$$D \approx \text{call setup dalay}$$

As shown in Figure 8, for $\rho_n = 2.84$, $b = 0.901$(without control) and $b = 0.123$(with control). So, we can roughly estimate the call setup delay

$$\begin{aligned} D &= 0.663 \text{ sec} && \text{with control} \\ &= 7.24 \text{ sec} && \text{without control} \end{aligned}$$

Although the above eatimated values are not so accurate, we can see that as the blocking probability increases the call setup delay increases.

Owing to a long call setup delay and a large blocking probability, user abandons a call and the call is forced to be cleared. Figure 11 shows user abandon rate and rate of SIP message retry out. User abandons a call when he does not hear a ring back tone in a long period. As ρ_n increases, the silent duration increases. So, user abandon rate increases as ρ_n increases. Thus, the throughput decreases. In addition, as the blocking probability of buffer increases, the rate of retry out of SIP messages increases. The retry out of SIP messages clears a call. So the throughput decreases if the overload control is not applied. Figure 11 shows that the overload control improves both of user abandon rate and rate of SIP message retry out.

As shown here, the proposed overload control improves the performance and protects from the throughput degradation.

V. Conclusion

The paper has proposed an overload control for a SIP signaling network. The proposed overload control has been evaluated using the network simulator (ns-2). The paper has used the throughput as the performance measure. Compared to the performance under the condition without control, the paper confirms that the overload control works well and the throughput characteristic can be improved.

In the future, we will shows how to find the most suitable control parameters, such as h, l and the value of Timer D.

References

[1] H. Schulzrinne and J. Rosenberg: "The Session Initiation Protocol: Internet-Centric Signaling", *IEEE Communication Magazine*, vol.38, 10, pp-134-141, Oct.(2000)
[2] J.Rosenberg, et.el.: "SIP: Session Initiation Protocol", *RFC3261*, http://www.ietf.org/rfc/rfc3261.txt, June(2002)
[3] M.Ohta: "Simulation study of SIP Signaling in an Overload Condition", *3rd Int'l Conf. on communications, Internet, and Information Technology*, pp.321-326, Nov.(2004)
[4] M. Govind, S. Sundaragopalan, Binu K S, and Subir Saha: "Retransmission in SIP over UDP - Traffic Engineering Issues", *Proc. of International Conference on Communication and Broadband Networking*, Bangalore, May(2003)
[5] J.H.James, B. Chen and L. Garrison: "Implementing VoIP: A Voice Transmission Performance Progress Report", *IEEE Communication Magazine*, vol.42, 6, pp.36-41, June(2004)
[6] Y. Xu, M. Westhead and F. Baker: "An Investigation of Multilevel Service Provision for Voice over IP Under Catastrophic Congestion", *IEEE Communication Magazine*, vol.42, 6, pp.94-100, June(2004)
[7] G. Camarillo, R. Kantola and H. Schulzrinne: "Evaluation of Transport Protocols for the Session Initiation Protocol", *IEEE Network*, Vol.17, 5,pp.40-46, Sep.(2003)
[8] T.Eyers and H. Schulzrinne: "Predicting Internet Telephony Call Setup Delay" *Proc. 1st IP-Telephony Wksp.*,Jan.(2000)
[9] M. Cortes, J. R. Ensor and J. O. Esteban:"On SIP Performance", Bell Labs Tech. J, 9, pp.155-172(2004)
[10] R.P.Ejzak, C.K.Florkey and R.W.Hemmeter: "Network Overload and Congestion: A Cpmparison of ISUP and SIP" *Bell Labs Technical Journal*, 9, pp.173-182(2004)
[11] VINT Project: "Network simulator ns-2", http://www.isi.edu/nsnam/ns/