Comparison of loss-based overload control mechanisms in signaling system with SIP protocol

Jan Rogowski, Halina Tarasiuk
Institute of Telecommunications
Warsaw University of Technology
Warsaw, Poland
j.m.rogowski@stud.elka.pw.edu.pl, halina@tele.pw.edu.pl

Abstract— **In this paper we compare two approaches for loss-based overload control mechanisms in signaling system with SIP protocol, hop-by-hop and end-to-end. Both discussed mechanisms are based on using a simple non-preemptive priority queueing scheme in SIP server. The effectiveness of both approaches is compared with the system with no overload control. The effectiveness of the solution is measured by average call setup time, goodput and basic fairness.**

Keywords-component; SIP protocol; signaling system; overload control mechanisms; basic fairness.

I. INTRODUCTION

Commonly known application signaling protocol, Session Initiation Protocol (SIP) [1], allows us to establish, modify or release connections e.g. for voice over IP applications. We expect that signaling system with SIP protocol should offer similar level of overload control as e.g. in PSTN [2]. Even under overload conditions, new calls should be still accepted and the call setup time should match requirements for PSTN networks. We propose a simple non-preemptive priority queuing scheme for to handle overload control messages with high priority while other SIP messages with low priority. By using such a scheme, SIP servers can avoid handling signaling messages, which will be dropped e.g. in next SIP server.

II. SIGNALING SYSTEM WITH SIP PROTOCOL

We consider the signaling system model with overload control mechanism as discussed in [3]. In this model each two SIP servers (sending entity and receiving entity) can be logically defined as follows: (1) SIP processor processes SIP messages and this component is protected from overload; (2) Monitor, control function and actuator are a part of overload control layer. In this scenario, monitor measures SIP processor occupancy and sends sample S to control function. Control function determines if overload has occurred and whether a throttle T needs to be set to reduce the load.

In the hop-by-hop implementation, monitor is located in SIP server and its associated actuator is in its direct downstream neighbor. Each SIP server throttles load that would otherwise overload its upstream neighbor. The end-to-end implementation takes advantage of multiple monitors providing overload control feedback to the actuator. The point is to throttle the traffic as close to an ingress server as possible. That allows to throttle traffic on the first SIP server and not to use server's resources to process messages that would be rejected.

For monitoring we use occupancy algorithm [4]. In each interval T_m monitor measures server's SIP processor occupancy. It is averaged by EWMA algorithm with weight w. Load reduce factor φ according to equation:

$$\varphi = \frac{\rho_{target}}{\rho_m} \quad , \tag{1}$$

is send to the actuator each T_f interval, where φ – load reduce factor, ρ_{target} – target utilization, ρ_m – measured utilization. When the actuator receives feedback it adjusts rejection factor p accordingly,

$$\mathrm{p} = \begin{cases} 1 & when\ p \cdot \varphi > 1 \\ p_{min} & when\ p \cdot \varphi < p_{min} \\ p \cdot \varphi & in\ other\ cases, \end{cases} \tag{2}$$

where p denotes INVITE rejection probability and p_{min} - minimum rejection probability.

III. RESULTS

Based on the discussed system model we have developed a discrete event simulator to evaluate performance metrics of SIP signaling system. We measure goodput defined as a total rate of calls per second (cps), which were terminated by users in the system (successful calls) and fairness. By fairness we mean basic fairness [3], which means that every request has the same chance of acceptance.

First, we investigate topology with network diameter D=3 in which overloaded server can be two hops away from ingress servers (see Figure 1).

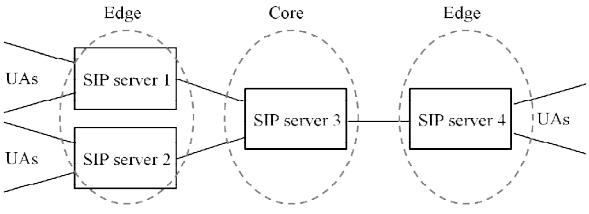

Figure 1. Network topology 1

User Agents (UAs) are connected to edge servers. Parameters for loss-based algorithm were set to T_f= 400 ms,

T_m = 200ms, w = 0.8, ρ_{target} = 0.95, ρ_{min} = 0.01. Edge servers have processing rate of 300 messages per second and rejecting rate (defined as rate at which server can generate 503 SIP messages) of 1200 messages per second while core server has processing rate of 480 messages per second and rejecting rate of 1500 messages per second.

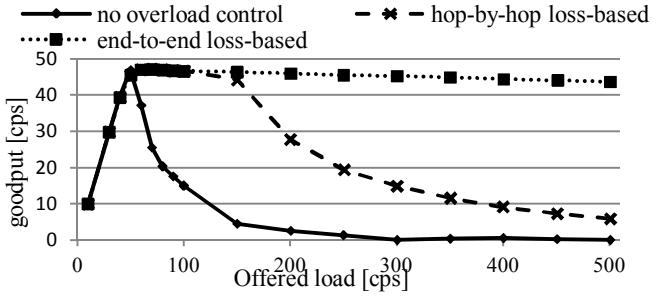

Figure 2. Goodput vs. offered load

Figure 2 shows that the system with no overload control mechanism becomes congested as soon as offered load reaches server call processing rate (approx. 50 calls per second) resulting in goodput drop. That is because servers 1 and 2 are not aware of overload of server 4 so they forward messages, which are going to be rejected anyway. Moreover, they processes server's 3 reject messages (503-service unavailable) which also increases CPU overhead. End-to-end mechanisms perform well at any load because new calls are rejected at ingress servers (as SIP server 1 and 2). Such approach, by better resource usage, provides the best results.

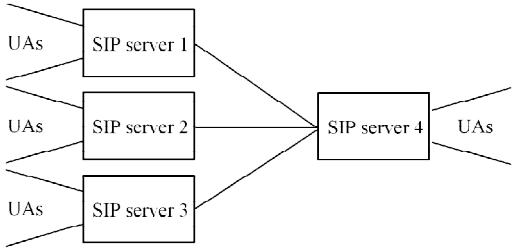

Figure 3. Network topology 2

Next, we investigate fairness properties of loss-based overload control mechanisms. Unlike other overload control mechanisms [3] loss-based does not need feedforward channel [5] to provide basic fairness. We assumed topology presented in Figure 3. We start with one server generating load and then we add consecutive servers as depicted in Figure 4. Figure 5 presents characteristic showing way in which resources of core SIP server 4 are shared between sending entities.

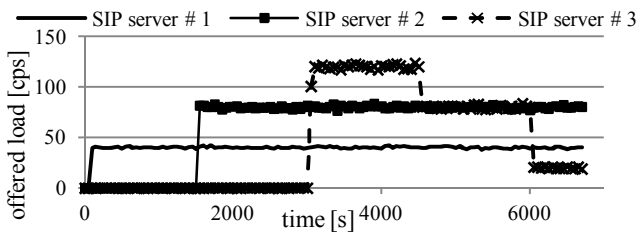

Figure 4. Offered load vs. time

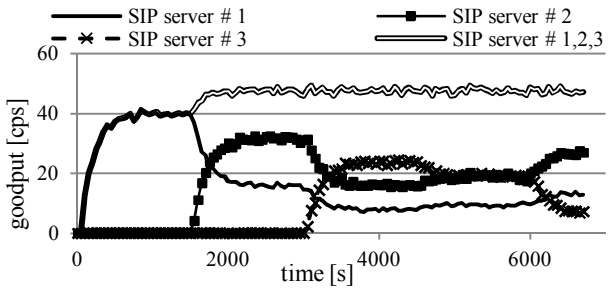

Figure 5. Goodput vs. time

In time interval from T=0 s to T=1500 s all resources are available for SIP server 1. In next time interval, from time T=1500 s to T=3000 s we have two sending entities, which generate load in 2:1 ratio (80 cps to 40 cps). One can observe that loss-based mechanism prevents overload and provides resources proportionally to the offered loads, that is, 32 cps to 16 cps. Such fairness property is basic fairness and is considered preferably in public networks environment. Server shares its resources proportionally to load offered by sending entities. Because of basic fairness, whatever the path is call request routed, larger streams have more resources assigned which results in the same call acceptance probability. Advantage of loss-based algorithm is that in order to provide such fairness it does not need feedforward channel required by other mechanisms as window-based or signal-based.

IV. SUMMARY

The paper provides performance evaluation of loss-based overload control mechanisms in the signaling system with SIP protocol. We have compared two approaches for overload control, hop-by-hop and end-to-end. We have presented the results showing better performance of end-to-end approach and we have also illustrated basic fairness for loss-based algorithm. Based on simulation results obtained for simple and complex topologies we conclude that signaling system with loss-based end-to-end overload control mechanism performs better than hop-by-hop and protects the system from overloading.

V. REFERENCES

[1] J. Rosenberg, et al.: "SIP: Session Initiation Protocol", Internet RFC 3261, June (2002)

[2] W. Berger: "Comparison of Call Gapping and Percent Blocking for Overload Control in Distributed Switching Systems and Telecommunications Networks". In: IEEE Transactions on Communications, vol. 39, no.4, April (1991).

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

[4] V. Hilt., I. Widjaja:" Controlling Overload in Networks of SIP Servers", In: 16th IEEE International Conference on Network Protocols, ICNP 2008, Orlando (2008)

[5] C. Shen, H. Schulzrinne, E. Nahura: "Session Initiation Protocol (SIP) Server Overload Control: Design and Evaluation", IPTComm 2008. LNCS, vol. 5310, pp. 149—173. Springer, Heidelberg (2008)