A Technique to Analyse Session Initiation Protocol Traffic

G. De Marco*
Department of Information and
Communication Engineering
Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-Higashi-ku, Fukuoka 811-0295,Japan
tel: +81-92-606-4970
e-mail: demarco@fit.ac.jp

G. Iacovoni
Ericsson Lab Italy
via Anagnina 203, Roma, Italy
e-mail:
giovanni.iacovoni@ericsson.com

L.Barolli
Department of Information and Communication Engineering
Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-Higashi-ku, Fukuoka 811-0295, Japan
tel: +81-92-606-4970
e-mail: barolli@fit.ac.jp

Abstract

The Session Initiation Protocol (SIP) will provide the signaling network in the next generation networks, such as UMTS and cdma2000. Here we study the traffic load generated by the use of this protocol from a methodological point of view. We propose an analytical technique to simulate SIP Finite State Machine (FSM) in an IP network by means of the theory of queuing networks. Based on this approach, we derive a simple model which can be applied to different network scenarios for performance evaluation and engineering purposes. We conclude this study by showing some results about call dropping rate in a wireless access network.

Keywords: SIP, FSM, closed queueing network, NS-2 simulator

1. Introduction

Complex functions of next generation networks can be carried out through signaling protocols and techniques whose performances are still under investigation. In particular the IP subsystem of 3G networks, known as IP Multimedia System (IMS), is conceived according to the functionalities of the SIP. This protocol arose in the IETF community as a general purpose protocol aimed at handling multimedia sessions in the Internet ([9]); subsequently, the 3GPP consortium adopted SIP as the signaling framework for Universal Mobile Telecommunication Networks (UMTS).

To the best knowledge of the authors there are few works on the performance analysis of SIP protocol from a traffic engineering point of view. For example, the impact of signaling load of the SIP for Telephony (SIP-T) is analyzed in [10], without considering all the states of the protocol. Mobility considerations are discussed in [6] and [8], where the hand-off delay is computed as a function of the total arrival rate at the base station, disregarding retransmissions and other type of messages. A first attempt to optimize the performance of signaling protocol has been addressed in [7], even though the authors focus only on authentication messages. However quality of service parameters, such as the mean call dropping rate, have never been quantified, although this kind of knowledge is crucial for an operator running the network.

In general, from a SIP signaling point of view, the communication system comprises a large number of parameters, such as the mobility pattern of the terminals, the statistics of signaling messages (e.g. the frequency of SIP REGISTER, or the probability of a SIP Not-Found message), the number and the type of hops traversed by a SIP message, and so on. Event-driven simulations do not scale well

* On leave from Department of Information Engineering and Electrical Engineering, University of Salerno, Italy.
This work is part of the PRESTO project, partly funded by Italian Ministry of University and Research (MIUR).

both with the number of connections injected in the simulation and with the complexity of the network scenario to be simulated.

In this paper, we use a queuing network approach to simulate a large population of SIP connections accessing an IP network. More precisely, we use the *closed queueing network* technique to model the Finite State Machine (FSM) of the SIP protocol according to the standard, see [9]. This technique has been applied to speed up simulation of TCP [4] in the case of many flows, the seminal work being [2]. Our underlying assumption is that SIP sessions can be modeled as customers of a queuing network. The queuing network is then coupled with the model for the IP network and a unique solution is attained via iterative methods. This approach shortens simulation times, yields fast capacity evaluation in realistic network scenarios. Furthermore, even though we apply this approach to a simplyfied network scenario, it provides the basis for performance analysis for more general cases, e.g. the entire set of SIP messages in a realistic IP path of IMS.

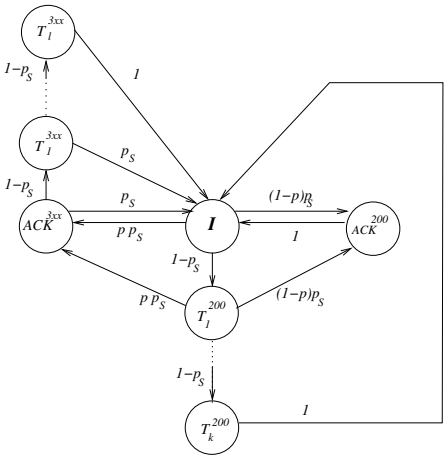


Figure 2. Queuing network of the SIP submodel

The structure of the paper is the following. In Section 2 we summarize the basic functionalities of SIP. Then, in Section 3 the proposed model is shown, along with some examples of applications. The paper ends with our conclusions in Section 4.

2. Signaling scenario with SIP

SIP is a transactional protocol at the application layer of the OSI stack[1, 9]: a fixed number of request/response messages are designed for a particular transaction. Essentially, there are two types of transactions, namely the INVITE and the non-INVITE transactions. INVITE transactions are initiated by the client transaction layer of a *SIP entity*. When a user wishes to communicate with an end host, her/his terminal sends an INVITE message to the SIP Proxy in her/his access network.

The SIP Proxy then sends back a provisional response, encoded in a status number, e.g. 100 TRYing. We consider that SIP messages are transferred through UDP. In this case the use of timers is mandatory to ensure a reliable communication.[1]

Two main situations can occur in response to the INVITE request, each of them ruled by its specific state machine. In the first one, Fig. 1-a, the correspondent user is available to establish a connection[2], while in the second one, Fig. 1-b, the correspondent user is not reachable (e.g. because its terminal is switched off) . Now let us examine the two cases above separately.

In the first case at every INVITE, the state machine starts Timer A, whose initial value is T_1. If no provisional response, i.e. 100TRYing or 180RINGing, or final response, i.e. 200OK, has been received, the timer value is doubled and a new INVITE is retransmitted, till a timeout value (*Timer B*) is reached, after which the transaction is terminated. If the User Agent (UA) client, sender side, receives a provisional response, it stops timer A and waits for a final response for a period of time lasting at maximum *Timer D*.

In the second case, a similar scheme is applied to Timer G (whose timeout is *Timer H*) to handle the retransmissions of final responses 3xx at the server side, as shown in Fig. 1-b.

As far as the correspondent user is concerned, Fig.1-c, *Timer* J is the timeout value for the UA client at correspondent side. When it expires no more final responses are sent[3].

3. Model of INVITE transactions

In this Section we describe the model along with its range of applicability.

First of all, we model an access network[4] where a fixed number M of SIP connections are present. The number M could change if, for example, a mobility pattern is taken into account. We assume that if a connection is closed after a timeout occurrence (e.g. *Timer B* or *Timer H* in Fig. 1), it

1 Our modelling approach is independent of the considered transport. The adaptation to a reliable transport is straightforward.

2 We assume that connection and session are synonyms.

3 Also non-INVITE transactions make use of timers, but in this preliminary study we concentrate only on a very simple situation.

4 For example, in the UMTS terminology, the access network could be represented by three paths: 1) the path from the user terminal to the Proxy- Call State Control Function (P-CSCF); the P-CSCF Interrogating/CSCF path and the I-CSCF/Serving-CSCF path.

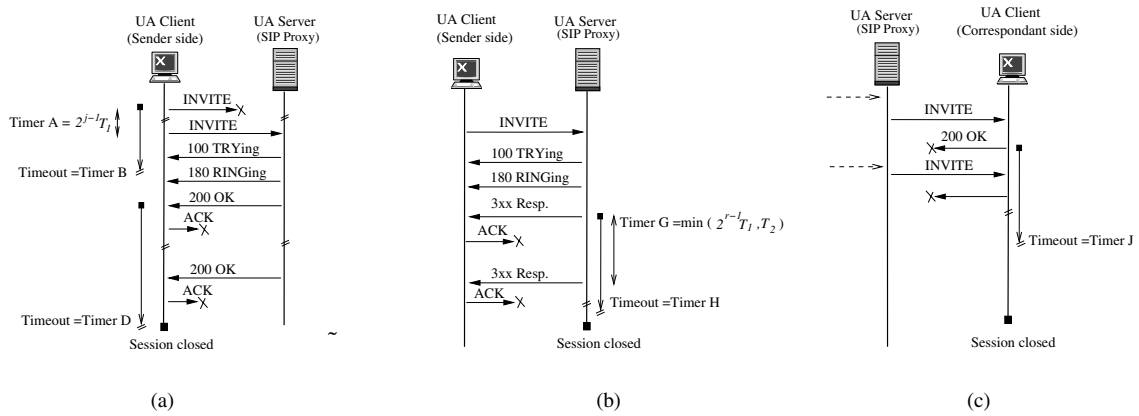


Figure 1. Timers in the SIP state machine: in a) Timer A is doubled at every retransmission of INVITE message and upper bounded by *Timer B*; in b) Timer G is associated to the final response 3xx, and upper bounded by *Timer H*; and in c) Timer J is started at every response from the correspondent user. UA = User Agent.

is re-opened immediately. If an INVITE$\longrightarrow$200OK transaction is successfully terminated, it is immediately re-opened as well.

The two basic approximations of our model are the following:

1. The reverse path (from the server to the terminals) is transparent, i.e. the are no delay and no losses.
2. The forward path is always heavily loaded. This results in a constant value for the round trip time (r_{tt}).

The first approximation could be accepted if we suppose that the reverse path is not as congested as the access network. The second approximation makes our model a conservative one. However, these assumptions give a very simple and tractable model of the IP path, as explained in [2].

The ratio of the model is the classical *divide and conquer* principle. The entire model is divided in two sub-models, one for the IP path and the other one for the FSM of SIP. The SIP sub-model is represented by a closed queuing network where the customers are the SIP sessions. The customers of a generic queue are sessions in the same state. The routing of the sessions inside the queueing network is governed by the transition probability p_S, i.e. the successful probability for the transmission of a packet along the IP path. The steady state distribution of the arrival rates into the queuing network is an input variable for the IP sub-model: given an arrival rate vector, the IP sub-model return a new p_S, which is fed again into the SIP sub-model. The model provides a unique solution in terms of p_S and distribution of arrival rate.

The procedure is therefore a fixed-point algorithm. We use the following notation:

- Q = the $N-tuple$ of the possible states of the SIP FSM;
- $\overrightarrow{\lambda} = (\lambda_i)_{i=1}^{N}$ the arrival rate vector, with λ_i being the arrival rate at the queue i ;
- $A = \{p_{ij}\}$ the routing matrix of the queueing network, e.g. the transition probability between the $i-th$ and $j-th$ queue;
- M = the fixed number of SIP sessions;
- C = the bottleneck (or available) transmission rate of the IP path, measured in bit/s;
- R = the total capacity of the IP path measured in packets of D bit;
- r_{tt} = the round trip time measured in $seconds$;
- p = the probability of a 3xx Response reception after an INVITE request.
- p_S = the successful probability of a packet transmission.

The above mentioned assumptions give $r_{tt} = \frac{DR}{C}$.

In the closed queueing network model, Q is the set of queues in the SIP sub-model.

With this notation, the SIP sub-model is represented by the vector equation $A\overrightarrow{\lambda} = b$,, where b is computed by means of a normalization equation. In Fig. 2 we show the pictorial representation of the SIP sub-model, the set of possible queues being $Q = (I, ACK^{200}, ACK^{3xx}, T_j^{200}, T_r^{3xx})$, with $1 \leq j \leq k$,

Q	$Description$	$Service\,Time$
I	INVITE sent	r_{tt}
ACK^{200}	200OK received, ACK sent	0
ACK^{3xx}	3xx received, ACK sent	0
T_j^{200}	Timer A expired	$2^{j-1}T_1$
T_r^{3xx}	Timer G expired	$2^{r-1}T_2$

Table 1. Description of the Queues of the model

$1 \leq r \leq l$, and k and l the maximum number of timer expirations for the INVITE request and ACK on 3xx Response, respectively. The description of the queues of Q is given in Table 1. For ease, the service time of T_r^{3xx} queues is assumed to be $2^{r-1}T_2$. It is worth noting that the timers T_r^{3xx} are on the SIP Proxy server side and indirectly induce traffic on the forward path (from terminals to the server).

The IP path sub-model we consider is a buffer always full, with a FIFO serving discipline of rate C. Assuming that a session in a queue produces a SIP packet of $D\,bit$, we have:

$$p_S = \frac{C}{D \sum_{1 \leq i \leq n} \lambda_i}, \tag{1}$$

where the summation is extended to those queues that produce traffic in the IP path . For example, the waiting state T_k^{200} is not inserted in (1). Thus, $n = (k-1) + (l-1) + 3$.

The solution of the model is the pair $(\vec{\lambda}, p_S)$. By means of the fixed-point procedure described above, the global solution (we suppose that it is unique, and it is always found) is reached when the difference between the results of two sub-models is less than a certain threshold, e.g. 10^{-65}.

The metric or the output of the model and the NS simulated network is the Call Dropping rate, P_{cd}. It is the probability that a session associated to a Call is dropped due to timeouts.

From the queuing network corresponding to the SIP submodel, it is not hard showing that:

$$P_{cd} = Prob\left\{ \left(I \rightarrow T_k^{200}\right) \bigcup \left(I \rightarrow T_l^{3xx}\right) \right\} = \\ = (1-p_S)^k + pp_S(1-p_S)^l \tag{2}$$

Example of Applications

5 To avoid oscillations around the fixed-point, we "filter" every solution by a gain which is lowered whenever the iterative procedure does not converge to the fixed-point

C	M	k	l	p	R	D
$10 \div 100Mbit/s$	$200 \div 5000$	7	6	0.2	200	$512bytes$

Table 2. Parameters for the Wired Scenario

3.1. Wired access

We perform simulations of the model with the parameters shown in Table 2. We simply assume that the size of a SIP packet is fixed for all SIP messages, even if SIP messages can be of various length , as it is for message with embedded multimedia information, like photos or audio/video clips, and for messages which had passed through several proxies. The maximum number of sessions injected in the system is 5000. The plot shown in Fig. 3 is the result of a simple routine written in MATLAB. For each point, the convergence is reached after a maximum of few hundreds of iterations, which corresponds to some minutes of simulation time on our machine, a Pentium IV 2,40 GHz 1Gbyte RAM.

3.2. Wireless/Wired access

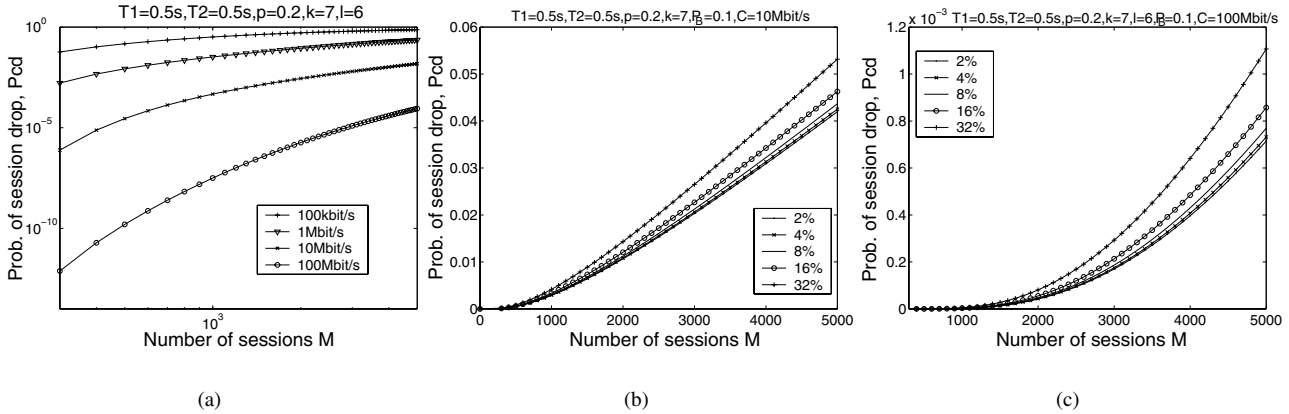
The model can be easily extended to other scenarios. In fact, let suppose that the SIP sessions come from a pool of radio accesses, in such a way that any SIP message traverses a radio link before entering the wired network, which is represented by the equivalent capacity equal to C. In this case, p_S must be replaced by the probability of a successful transmission $1 - P_L$, P_L being the packet loss given by:

$$P_L = (1-p_S) + P_w - P_w(1-p_S) \tag{3}$$

where P_w is the loss probability due to radio link and p_S is still computed as in Eq. 1. To compute P_w we use the well known two state Gilbert-Elliot model which is a good approximation for the packet loss probability over a radio link affected by fading phenomena ([3]). In particular, if we call the steady state probabilities of the *good* state and *bad* state with π_G and π_B respectively, we obtain from Eq. 3:

$$P_L = (1-p_S)\left(\pi_G(1-P_G) + \pi_B(1-P_B)\right) \quad + \\ + \pi_G P_G + \pi_B P_B$$

where P_G and P_B are the packet loss probabilities corresponding to *good* and *bad* state, respectively. The packet loss P_L is then injected in the SIP sub-model, described by the Eq. **??** and the system is analyzed as explained in Section 3. In Fig. **??**, we show P_{cd} versus M for different values of P_G and C. Interestingly, for $P_G < 8\%$, the Call Dropping Rate is independent of P_G, since in this parameter set-



(a) (b) (c)

Figure 3. Call dropping rate for the case of wired (a) and wireless (b)(c) access network as a function of the equivalent bottleneck capacity C. In (b), $C = 10Mbps$, in (c) $C = 100Mbps$

tings losses mainly occur in wired access network. However, P_{cd} values which can be accepted in operating networks lie between 10^{-2} and 10^{-3} as pointed out in [5]. We would emphasize that these network parameters have chosen for a validation purpose only, and they do not necessarly represent values found into the real scenario. Further sudies will extend the methodology to more exhaustive cases, as well as the analysis of SIP model covering other kind of signaling messages and network architectures more realistic.

4. Conclusions

In this paper we gave a general criterion to analyze signaling networks. We focused on how to build the model of the SIP protocol and the IP network, with two levels of complexity: one for wired access and the other one for wireless access. The two levels are clearly not exhaustive, since we made some simplistic assumptions, both for the SIP protocol and the IP network. However, these simplifi cations have been made as a fi rst step towards more complicated scenarios. This model allows a fast capacity evaluation of the signaling network for engineering applications. We are now studying a more complete FSM of SIP protocol which examines non-INVITE transactions, variable length of packets and various mobility patterns. These extensions to the present work could be easily achieved by introducing multiple classes of customers in the queuing network model of the system.

References

[1] IETF Sip Working Group. http://www.ietf.org/html.charters/sip-charter.html.

[2] R. L. Cigno and M. Gerla. Modeling window based congestion control protocols with many flows. In *Performance Evaluation*, volume 36-37, pages 289–306. Elsevier Science, 1999.

[3] R. Fracchia, M. Garetto, and R. L. Cigno. A queuing network model of short-lived tcp flows with mixed wired and wireless access links. In *Proceedings of Second International Workshop on QoS-IP 2003*, pages 392–403, feb 2003.

[4] M. Garetto, R. Cigno, M.Meo, and M. A. Marsan. A detailed and accurate closed queuing network model of many interacting tcp flows. In *Proc. IEEE INFOCOM'01*, pages 1706–1715, apr 2001.

[5] ITU-T Q.543. *Digital Exchange Performance Design Objectives*. International Telecommunication Union, 1994.

[6] K.D.Wong, A.Dutta, J.Burns, R.Jain, K.Young, and H.Shulzrinne. A multilayered mobility management scheme for auto-confi gured wireless ip networks. *IEEE Trans. Wireless Commun.*, 10:62–99, Oct. 2003.

[7] Y.-B. Lin and Y.-K.Chen. Reducing authentication signaling traffi c in third-generation mobile network. *IEEE Trans. Wireless Commun.*, 2(3):493–501, may 2003.

[8] N.Banerjee, K.Basu, and S.K.Das. Hand-off delay analysis in sip-based mobility management in wireless networks. In *Proc. IEEE PDS'03*, apr 2003.

[9] J. Rosemberg et al. Sip:session initiation protocol, jun 2002. IETF RFC3261, http://www.ietf.org/rfc/rfc3261.txt.

[10] J. Wu. and P. Wang. The performance analysis of sip-t signaling system in carrier class voip network. In *Proc. IEEE AINA'03*, volume 2, pages 39–44, mar 2003.