

# A Theoretical Model of the Effects of Losses and Delays on the Performance of SIP

Dorgham Sisalem  
Tekelec  
Berlin, Germany  
dorgham.sisalem@tekelec.com

Mikkel Liisberg  
Tekelec  
Berlin, Germany  
mikkel.liisberg@tekelec.com

Yacine Rebahi  
Fraunhofer Fokus  
Berlin, Germany  
rebahi@fokus.fraunhofer.de

**Abstract**—The Session Initiation Protocol (SIP) is widely used for VoIP communication. Losses caused by network or server overload would cause retransmissions and delays in the session establishment and would hence reduce the perceived service quality of the users. In order to be able to take counter measures network and service planners require detailed models that would allow them to predict such effects in advance. This paper presents a theoretical model of SIP that can be used for determining various parameters such as the delay and the number of messages required for establishing a SIP session when taking losses and delays into account. The model is restricted to the case when SIP is transported over UDP. The theoretical results are then verified using measurements.

## I. INTRODUCTION AND MOTIVATION

The session initiation protocol (SIP) is increasingly becoming the de-facto standard for VoIP deployments in fixed and wireless networks. SIP can be used over various transport protocols such as UDP, TCP or SCTP. To enable the reliable transmission of SIP messages even when used over UDP, SIP supports application level retransmission mechanisms. That is in case no response was received for a sent request then after a timeout the request is retransmitted. Thereby, losses due to overloaded servers or lossy links would cause delays in the session establishment and hence reduce the perceived service quality.

In this paper we provide a theoretical model that can be used by operators and network designers to determine the effects of introducing a SIP-based service to their networks in terms of bandwidth usage for example and the effects of losses and delays on the service quality. This model uses as the input various traffic characteristics such as the number of calls per second and mean holding time and network characteristics, such as losses and propagation delays. The output of the model provides details on the bandwidth and delay needed for successfully establishing a session when using SIP over UDP. The effects of using a reliable transport protocol such as TCP or SCTP are out of scope of this paper and are left for future work.

In Sec. II we provide a short introduction to SIP and present a brief overview of the literature concerning modeling of SIP.

In Sec. III the SIP model is presented. Measurements verifying the correctness of the model are then presented in Sec. IV. In Sec. V some of the limitations of the model are described and suggestions for further refinements are discussed.

## II. BACKGROUND AND RELATED WORK

In general a SIP-based VoIP service, see [1], consists of user agents (UA), proxies and registrar servers. The UA can be the VoIP application used by the user, e.g., the VoIP phone or software application, a VoIP gateway which enables VoIP users to communicate with users in the public switched network (PSTN) or an application server, e.g., multi-party conferencing server or a voicemail server.

The registrar server maintains a location database that binds the users' VoIP addresses to their current IP addresses.

The proxy provides the routing logic of the VoIP service. When a proxy receives a SIP request from a user agent or another proxy it also conducts service specific logic, such as checking the user's profile and whether the user is allowed to use the requested services. The proxy then either forwards the request to another proxy or to another user agent or rejects the request by sending a negative response.

With regard to the SIP messages we distinguish between requests and responses. A request indicates the user's wish to start a session (INVITE request) or terminate a session (BYE request). We further distinguish between session initiating requests and in-dialog requests. The INVITE request used to establish a session between two users is a session initiating request. The BYE sent for terminating this session would be an in-dialog request. Responses can either be final or provisional. Final responses can indicate that a request was successfully received and processed by the destination. Alternatively, a final response can indicate that the request could not be processed by the destination or by some proxy in between or that the session could not be established for some reason. Provisional responses indicate that the session establishment is in progress, e.g, the destination phone is ringing but the user did not pickup the phone yet.

A SIP proxy acts in either stateful or stateless mode. In the stateful mode, the proxy forwards an incoming request to its

destination and keeps state information about the forwarded request until either a response is received for this request or a timer expires. When used over an unreliable transport protocol such as UDP, if the proxy did not receive a response after some time, it will resend the request. In the stateless mode, the proxy would forward the request without maintaining any state information. In this case the user agent would be responsible for retransmitting the request if no responses were received.

SIP uses an exponential retransmission behavior. So if a sender of a SIP message does not receive a response after some time, it will resend the request after some waiting time. In case no response was received for the retransmission, the sender increases the waiting time and tries again and so up to a certain number of retransmissions, see [1].

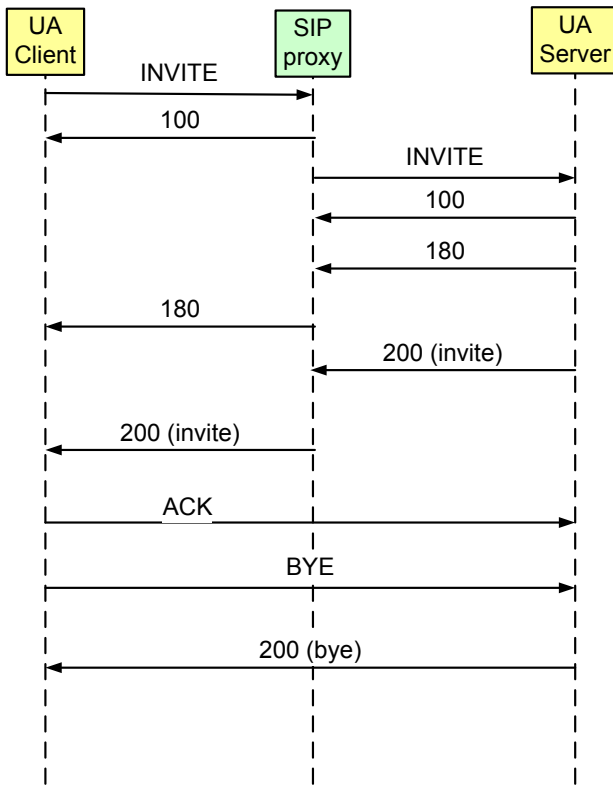


Fig. 1. SIP session signaling

Fig. 1 depicts a typical SIP session as usually used for benchmarking the performance of SIP implementations, see [2]. The session establishment is triggered by the sending of an INVITE request. In general, the session establishment can be considered as consisting of three phases, namely:

- INVITE phase: This phase is initiated with the sending of an INVITE request and is terminated when the client receives a provisional or final response. This phase actually consists of two parts, the communication between the UA client and the SIP proxy and the communication

between the SIP proxy and the UA server. Once a stateful SIP proxy receives the INVITE request from the client it replies with a provisional response, e.g., 100 or 180, to the client. In case the INVITE or the provisional response are lost, the client will retransmit the request after the expiration of a timeout value called T1. After receiving the INVITE, the SIP proxy forwards the request to the UA server which would reply with a provisional or final response. In case either the request or all of the responses were lost then the proxy would retransmit the request after T1 seconds.

- Final response phase: This phase is started when the UA server responds with a final positive or negative response and terminated when the server receives an ACK message from the client. That is, if either the response or the ACK were lost the UA server would retransmit the response after T1 seconds. Note that the retransmission is done here on an end-to-end basis and not hop-by-hop.
- BYE phase: This phase is initiated with the sending of a BYE request and is terminated when the sender of the BYE receives a response to it. In case the BYE or the response were lost then the sender of the BYE would retransmit it. Similar to the final response phase the retransmission is done here on an end-to-end basis.

With the success of SIP, there have been a limited number of studies addressing aspects of performance evaluation and modeling of SIP. Chebbo et al. describe in [3] a modeling tool with which it is possible to estimate the number of required SIP entities for supporting certain traffic. Gurbani et al. present in [4] a theoretical model of a SIP server using queuing theory. This model is then used to evaluate the performance of a SIP server in terms of response time and number of served requests. Wu et al. analyse in [5] the usage of SIP for carrying telephony information in terms of queuing delay and delay variations. In general, these studies aim at investigating the performance of SIP servers in terms of the number of SIP sessions that can be supported by a SIP server or the processing delays at such servers. In contrast, in our work we do not aim at modeling the performance of a SIP server but to investigate the performance of SIP in terms of number of messages and amount of time needed by SIP for establishing a session in lossy environments.

### III. SIP PERFORMANCE MODEL

Fig. 2 depicts an often used VoIP topology. This infrastructure consists of SIP user agent clients that initiate calls to user agent servers. The calls traverses two SIP proxies. Each link of the depicted network has a loss rate of  $l$  and has a propagation delay of  $D$  seconds. Each proxy has a capacity of  $C$ , e.g. it can support  $C$  requests per second.

When used with UDP, SIP uses an exponential retransmission behavior to enable reliable message delivery. When a SIP sender does not receive a response to a sent request, either

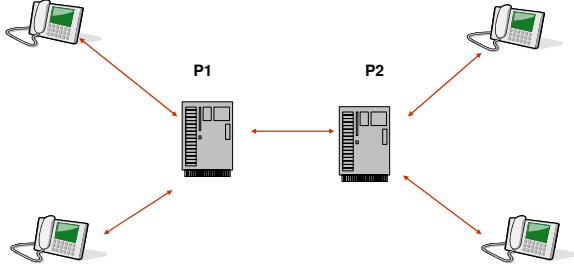


Fig. 2. Test environment

because the request itself or the response to it was lost, the sender retransmits the request after  $T1$  seconds, see [1], and the timeout value is increased. After a maximum number of retransmissions were sent the sender will give up. For the case of INVITE requests, the exponential retransmission behavior is used up to the so called TimerB. That is a request is retransmitted at time points  $T1$ ,  $3T1$ ,  $7T1$ ,  $15T1$  and up to TimerB. This can be represented as a series in the form of:

$$(2^1 - 1)T1, (2^2 - 1)T1, (2^3 - 1)T1, \dots, (2^N - 1)T1 \quad (1)$$

with  $((2^N - 1)T1 = \text{TimerB})$ . Thereby the maximum number of retransmitted INVITE requests ( $N_i$ ) is

$$N_i = \left\lfloor \frac{\ln\left(\frac{\text{TimerB}}{T1} + 1\right)}{\ln(2)} \right\rfloor \quad (2)$$

Non-INVITE requests use the exponential retransmission behavior up to a timer called  $T2$  and then every  $T2$  seconds up to the so called TimerF. That is a request is retransmitted at time points  $T1$ ,  $3T1$ ,  $7T1$ ,  $15T1$  and up to  $T2$ . Then at  $2T2$ ,  $3T2$ , and up to TimerF. This can be represented as a series in the form of:

$$(2^1 - 1)T1, (2^2 - 1)T1, (2^3 - 1)T1, \dots, (2^{N_o} - 1)T1, \\ 2(2^{N_o} - 1)T1 \dots \text{TimerF} \quad (3)$$

with  $((2^{N_o} - 1)T1 = T2)$ . The number of retransmissions ( $N_o^e$ ) conducted in the exponential manner up to the  $T2$  timer is determined as:

$$N_o^e = \left\lfloor \frac{\ln\left(\frac{T2}{T1} + 1\right)}{\ln(2)} \right\rfloor \quad (4)$$

After  $T2$  seconds, the retransmission timeout is kept constant to  $T2$ . The maximum number of retransmissions of a non-

INVITE request ( $N_o$ ) can then be determined as:

$$N_o = N_o^e + \left\lfloor \frac{\text{TimerF} - (2^{N_o^e} - 1) \times T1}{T2} \right\rfloor \quad (5)$$

with  $((2^{N_o^e} - 1) \times T1)$  indicating the time point at which the sender goes from exponential backup to a constant timeout value.

#### A. Bandwidth Consumption of SIP Signaling for Lossy Networks

With a loss rate of  $l$ , out of  $r$  issued INVITE requests per  $T1$  seconds ( $r \times l$ ) packets would be lost on average. These would be retransmitted  $T1$  seconds later. The retransmitted packets would also suffer from a loss and will have to be retransmitted later. Hence, the call generation rate ( $R_i$ ) can be depicted is shown in Tab I.

Time	$R_i$	lost
0	$r$	$lr$
1 T1	$r + lr$	$lr + l^2r$
2 T1	$r + lr$	$lr + l^2r$
3 T1	$r + lr + l^2r$	$lr + l^2r + l^3r$
4 T1	$r + lr + l^2r$	$lr + l^2r + l^3r$
5 T1	$r + lr + l^2r$	$lr + l^2r + l^3r$
...	...	...
7 T1	$r + lr + l^2r + l^3r$	$lr + l^2r + l^3r + l^4r$
8 T1	$r + lr + l^2r + l^3r$	$lr + l^2r + l^3r + l^4r$
...	...	...

TABLE I  
RETRANSMISSION BEHAVIOR OF INVITE REQUESTS DUE TO NETWORK LOSSES

At time point 0,  $r$  requests are sent per  $T1$  seconds. After  $T1$  seconds the senders will continue generating  $r$  new INVITE requests per  $T1$  second and will retransmit the lost ( $l \times r$ ) requests, e.g. ( $r + (l \times r)$ ) will be sent. Out of those ( $l \times (r + (l \times r))$ ) will be lost. These would be retransmitted at time  $3T1$ . At time  $2T1$   $r$  new requests will be sent plus the requests that were lost  $T1$  seconds ago, e.g., ( $l \times r$ ) requests. Out of the sent request ( $l \times (r + (l \times r))$ ) will be lost. These would be retransmitted at time  $4T1$  and so on.

The number of INVITE requests ( $R_i$ ) sent by the sender at any time point ( $n$ ) can, hence, be determined as:

$$R_i(l, n) = r \times \left(1 + \sum_{m=1}^{m=k} l^m\right) \quad (6)$$

with  $(k = \left\lfloor \frac{\ln(n+1)}{\ln(2)} \right\rfloor)$ .  $n$  can be maximally  $(\frac{\text{TimerB}}{T1})$ . At this stage the number of new losses, e.g. losses of newly generated requests, would become equal to the number of retransmissions that will be terminated as the maximum number of attempts was already tried. Hence, at this stage the system reaches a steady state.

For non-INVITE requests, the transmission behavior is slightly different, see Tab, II with the value of  $T2$  set to 4

Time	$R_o$	Lost
0	$r$	$lr$
1 T1	$r + lr$	$lr + l^2r$
2 T1	$r + lr$	$lr + l^2r$
3 T1	$r + lr + l^2r$	$lr + l^2r + l^3r$
4 T1	$r + lr + l^2r$	$lr + l^2r + l^3r$
5 T1	$r + lr + l^2r$	$lr + l^2r + l^3r$
....	....	....
7 T1	$r + lr + l^2r + l^3r$	$lr + l^2r + l^3r + l^4r$
8 T1	$r + lr + l^2r + l^3r$	$lr + l^2r + l^3r + l^4r$
....	....	....
15 T1	$r + lr + l^2r + l^3r + l^4r$	$lr + l^2r + l^3r + l^4r + l^5r$
16 T1	$r + lr + l^2r + l^3r + l^4r$	$lr + l^2r + l^3r + l^4r + l^5r$
....	....	....
23 T1	$r + lr + l^2r + \dots + l^5r$	$lr + l^2r + \dots + l^6r$
24 T1	$r + lr + l^2r + \dots + l^5r$	$lr + l^2r + \dots + l^6r$
....	....	....
31 T1	$r + lr + l^2r + \dots + l^6r$	$lr + l^2r + \dots + l^7r$
....	....	....

TABLE II  
RETRANSMISSION BEHAVIOR OF NON-INVITE REQUESTS DUE TO LOSSES  
IN THE NETWORK

seconds and T1 set to 0.5 seconds. In this case, the number of non-INVITE requests ( $R_o$ ) sent by the sender at any time point ( $n$ ) can be determined as:

$$R_o(l, n) = \begin{cases} r \times (1 + \sum_{m=1}^{m=k} l^m) & n \leq \frac{T_2}{T_1} \\ r \times (1 + \sum_{m=1}^{m=k} l^m + \sum_{m=k+1}^{m=q} l^m) & otherwise \end{cases} \quad (7)$$

with ( $k = \lfloor \frac{\ln(n+1)}{\ln(2)} \rfloor$ ) while ( $n \leq \frac{T_2}{T_1}$ ), e.g.,  $k$  would be maximally equal to  $N_o^e$ .  $q = \lfloor \frac{(n-2^{N_o^e}-1)T_1}{T_2} \rfloor$  which ensures that  $q$  is incremented every  $T_2$  seconds. Note that the maximum value of  $n$  here is ( $n = \frac{TimerF}{T_1}$ ) at which stage the steady state is reached, e.g., number of new retransmissions equals the number of terminated retransmissions.

1) *Losses During Invitation Phase:* As already described, an INVITE is sent reliably on a hop-by-hop basis. Hence if the INVITE sent by the UA client or the 100 response sent by the first proxy (**P1**) request were lost then the UA client would retransmit the INVITE. The same applies between the two proxies and between the second proxy (**P2**) and the UA server. Hence, we can consider the three hops as independent from each other. For each hop a request is considered to be successfully sent if the request and its response arrive at their destinations successfully. Thereby, one needs to consider the losses in both direction, e.g., an end-to-end loss ( $l_e$ ) of

$$l_e = 1 - (1 - l)^2 \quad (8)$$

The number of INVITE requests needed for setting up a SIP session ( $P_i$ ) can be determined as the number of messages sent after reaching the steady state divided by the number of original, e.g., not retransmitted, messages sent. So for example

if in the steady state 10 messages are sent per second where as the transmission rate is 5 messages per second then it would mean that on average 2 messages have to be sent to get one through.

$$P_i = R_i(l_e, N)/r \quad (9)$$

with  $N = \frac{TimerB}{T_1}$ .

The transmission of responses will only be triggered after the reception of an INVITE. With a one way delay of  $l$  and a steady transmission rate of  $R_i(l_e, N)$ , only ( $R_i(l_e, N) \times l$ ) INVITE requests can be received. Hence the number of responses sent during a session establishment  $P_{100}$  can be determined as

$$P_{100} = P_i \times l \quad (10)$$

Thereby, for successfully sending an INVITE request across the three hops, on the average ( $3 \times P_i + 3 \times P_{100}$ ) SIP messages would have to be transmitted. Note, however, that this is kind of a worse case calculation. So for example if the INVITE request from the first proxy (**P1**) was received by the second (**P2**) but the 100 response sent by **P2** was lost, then **P1** would resend the INVITE request after T1 seconds. However, if the INVITE request was sent successfully from **P2** to the UA server and the UA server sent back a 180 response before T1 expires, **P2** would forward the 180 response to **P1**. If the 180 response reaches **P1** then **P1** would suppress the retransmission of the INVITE.

2) *Final Response and ACK Phases:* After sending a final response, the UA server expects to receive a reply, e.g., an ACK, before T1 seconds. Hence, the relation between the final response and the ACK is similar to that between an INVITE and a provisional response. Unlike the INVITE requests, the relation between the final response and the ACK is an end-to-end one, e.g., the proxies in between would not retransmit the lost messages. Hence for determining the number of final response ( $P_f$ ) and ACK requests ( $P_a$ ) needed on the average for setting up a session, one can use the same equations as previously but by taking into account the end-to-end delay. Assuming that all requests follow the same path, e.g., the proxies record-route themselves in the SIP requests and with a loss rate of  $l$  on each link the one way end to end loss ( $L$ ) of a request traversing the three links would be

$$L = 1 - (1 - l)^3 \quad (11)$$

The end-to-end loss for a request plus response would in this case be

$$L_e = 1 - (1 - L)^2 \quad (12)$$

Similar to the determination of the number of INVITE requests, we can now determine the number of final responses, ACKs that are sent on average as follows.

$$P_f = R_o(L_e, N)/r \quad (13)$$

$$P_a = P_f \times L \quad (14)$$

The same discussion applies to the relation between the BYE requests and their responses. Hence, the number of BYEs ( $P_b$ ) and their responses ( $P_{200}$ ) that are needed on average for terminating a SIP session can be determined as follows:

$$P_b = R_o(L_e, N)/r \quad (15)$$

$$P_{200} = R_o(L, N)/r \quad (16)$$

3) *Total Bandwidth Usage*: The total amount of bandwidth ( $B$ ) that would be caused by the SIP signaling for a call establishment rate of  $r$  and an average SIP packet size of  $S$  is

$$B = r \times S \times (3 \times P_i + 3 \times P_{100} + P_f + P_a + P_b + P_{200}) \quad (17)$$

### B. Estimation of the Session Establishment Delay

In this section we estimate the session establishment delay when network losses cause the retransmission of SIP requests. The delay is determined as the time needed for completing both the INVITE and final response phases. The final response phase is conducted in an end-to-end manner. Thereby, the delay incurred by this phase would be equal to the delay caused by the round trip propagation delay ( $6D$ ) plus the time between the retransmissions. Thereby the average delay for completing the final phase ( $T_f$ ) can be estimated as:

$$T_f = 6D + T1 \times (2^{N_o^e - 1} - 1) + T2 \times (P_f - N_o^e) \quad (18)$$

with the first term of the equation describing the round trip delay, the second the delay caused by the retransmission during the exponential backoff phase and the last describing the delay when retransmitting the request every  $T2$  seconds. Note that the loss of the response messages would result in the retransmission of the request, and hence any additional delay caused by the loss is already accounted for in the increased number of request retransmissions. Hence, sending of the response only adds propagation delay to the total delay.

The INVITE requests can be retransmitted by each component traversed by the request. However, each component does not wait till the previous component has successfully received a response. For example the UA client sends an INVITE to **P1**. **P1** receives the INVITE and sends back a 100 response. If the 100 response is lost, the UA client would retransmit the request after  $T1$  seconds. However, **P1** does not need to consider this and would forward the INVITE to **P2** and start its own retransmission timeout. Hence, for determining the delay of the INVITE requests, one needs to consider the one-way loss ( $l$ ) between the hops. Thereby, the delay of the INVITE phase ( $T_i$ ) on the hops between the UA client and **P1** as well as **P1** and **P2** can be estimated as follows:

$$T_i = D + T1 \times \left(2^{\frac{R_i(l, N)}{r} - 1} - 1\right) \quad (19)$$

with  $R_i$  is determined as in Eqn. 9 with  $N = \frac{TimerB}{T1}$ . Note that once the INVITE request reaches the UA server

INVITE/Sec	1500	2000	2500
5%	1580	2130	2650
10%	1670	2260	2800

TABLE III  
THEORETICAL RESULTS FOR NETWORKS WITH LOSSES OF 5% AND 10%

the INVITE phase can be considered as finished as the UA server would start sending responses and would hence go into the final response phase.

Thereby the average total session establishment delay is then determined as the sum of ( $3T_i + T_f$ ).

## IV. MEASUREMENT BASED EVALUATION

To verify the correctness of the presented model, we conducted a set of measurements in a testbed resembling the one presented in Fig. 2. As the SIP proxy an open source implementation was used<sup>1</sup>. TimerB was set to 32 seconds, timer T1 to 0.5 seconds and timer T2 to 4 seconds, as recommended in [1].

For the tests, a number of senders generated INVITE requests to a number of receivers. For the lossy network measurements, a random loss generator was introduced between the two proxies. That is the loss rate on the links between the UA client and **P1** as well as between **P2** and the UA server was set to 0. The first SIP proxy (**P1**) is capable of accepting all incoming messages and replying with a provisional response to them. **P1** is responsible for retransmitting any lost requests.

Fig. 3 and Fig. 4 depict the measured number of INVITE messages when initiating SIP calls with different rates in a network with a loss rate ( $L$ ) of 5% and 10%. The transmission rate started with 1 call per second and was increased gradually by 250 additional calls each 5 seconds. Tab. III describes the

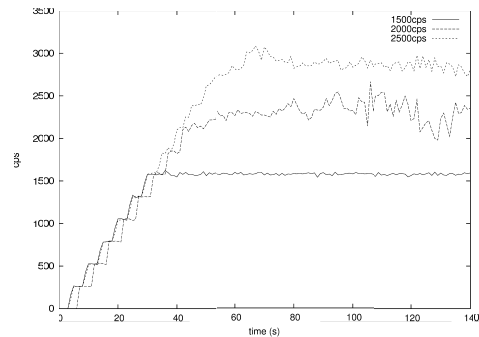


Fig. 3. Transmission behavior with a 5% loss probability

<sup>1</sup>See [www.iptel.org](http://www.iptel.org)

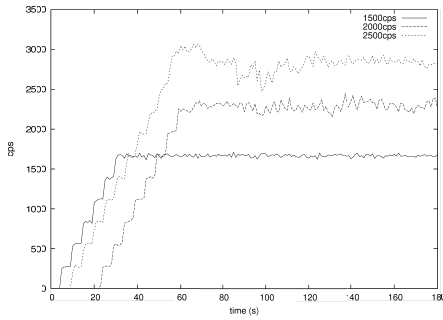


Fig. 4. Transmission behavior with a 10% loss probability

theoretical results using the presented model. So in the case of a call generation rate of ( $r = 1500$ ) one would end up having 1580 INVITE requests per second due to the retransmissions. Comparing between the results as depicted in Fig. 3 and Fig. 4 with the estimated results presented in Tab. III we can see that the deviation between the two is rather minor and stems mainly from measurement inaccuracy.

## V. SUMMARY AND FUTURE WORK

In this paper we presented a theoretical model for evaluating the performance of SIP signaling in lossy environments. Using this model various parameters were calculated such as the bandwidth needed for the signaling messages and the session establishment delay. The presented theoretical model was then verified using measurements.

While the measurement results are very promising the models will need to be verified more thoroughly using more elaborated loss models, different usage scenarios, traffic models as well as investigate the effects of overloaded servers. Further, the presented models assume using UDP as the transport protocol and do not consider TCP. As TCP uses its own retransmission mechanisms, the number of sent messages and the delay would differ from the results presented here.

With SIP gaining in complexity the message sizes are increasing as well leading to possible message fragmentation. This can further change the loss characteristics and needs to be considered in our future work. Additionally, we only considered calls to single destinations. SIP supports so called forking that would allow a proxy to forward a request to different destinations and wait for their responses. This would add additional complexities in determining the loss behavior and delay values.

Another aspect that needs to be considered is the performance of SIP in IMS networks [6]. Call models in terms of the used messages and message sequences are different in IMS from the used model here. A more complete model should take these difference into account.

## REFERENCES

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Jun. 2002.
- [2] H. Schulzrinne, S. Narayanan, J. Lennox, , and M. Doyle, "Sipstone - benchmarking sip server performance," Apr. 2002.
- [3] H. Chebbo and W. M., "Traffic and load modelling of an IP mobile network," in *Fourth International Conference on 3G Mobile Communication Technologies*, London, UK, June 2003.
- [4] V. K. Gurbani, L. Jagadeesan, and V. B. Mendiratta, "Characterizing session initiation protocol (SIP) network performance and reliability." in *ISAS*, ser. Lecture Notes in Computer Science, M. Malek, E. Nett, and N. Suri, Eds. Springer, pp. 196–211.
- [5] J.-S. Wu and P.-Y. Wang, "The performance analysis of SIP-T signaling system in carrier class voip network," in *AINA '03: Proceedings of the 17th International Conference on Advanced Information Networking and Applications*. Washington, DC, USA: IEEE Computer Society, 2003, p. 39.
- [6] "Signaling flows for the IP multimedia call control based on session initiation protocol (SIP) and session description protocol (SDP)," 3rd Generation Partnership Project, Technical Specification Group Core Network and Terminals, 2007.