

Performance Analysis of SIP as Signaling in Next Generation Network¹

Yin Zeming

State Key Laboratory of Switching Technology and Telecommunication Networks, Beijing University of Posts and Telecommunications, Box 187, Beijing 100876, China

yinzeming2002@163.com

Liu Yuzhang

liuyz@mail.dascom.com.cn

Yang Fangchun

fcyang@bupt.edu.cn

Abstract: Session Initiation Protocol (SIP) has been widely used for Signaling for VoIP networks now. In the coming 3rd Generation Mobile Network and even Next Generation Network, SIP plays an important role in setting up calls. SIP's performance is analyzed with two models in this paper, a coarse-granularity transaction-based model and a fine-granularity model considering congestion avoidance problem. The former model is general, and the later is based on the condition that TCP is the transport protocol to carry SIP. With the transaction-based model we can compare the difference of SIP's behavior over different physical links. With the fine-granularity model, the more concrete delay considering congestion avoidance problem can be got. A mechanism to ameliorate SIP's performance over connection-oriented transport protocol is analyzed with the models.

Key Words: SIP, delay, NGN, TCP, UDP, SCTP, slow start, transaction

1. Introduction

SIP is an application-layer control protocol that can establish, modify and terminate sessions, and it supports various facets of telecommunication such as user location, user capabilities, user presence, call setup and call handling [9]. Now SIP has been widely used as the signaling protocol in charge of call control in VoIP networks together with H.323. Although H.323 is more popular for Internet telephones, SIP will surely be the dominant signaling protocol in the future networks for its simplicity and scalability. In 3GPP R5, SIP has been appointed as the exclusive signaling to control the IP Multimedia Sub-system (IMS). In NGN 2004 (ITU's research project on NGN), SIP and its extension versions are adopted to interconnect the Softswitches or between an Application Server and a Softswitch as a call control signaling. The capability and performance of signaling protocols affect the effect of services that customers have ordered. Thus besides the research on SIP's capability, the performance analysis of SIP in new environments is necessary. Among the performance parameters for call setup, delay should be emphasized because it affects the users directly.

¹ The work is supported by National Nature Science Fund (90104024), National Science Fund for Distinguished Young Scholars (60125101)

users directly.

SIP can be transmitted over several transport protocols no matter they are connection-oriented or connectionless. The former with reliable transport mechanism and congestion avoidance mechanism such as TCP and SCTP utilize the network friendly and provide reliable transport services. UDP, one kind of the latter, has not reliable transport mechanism or congestion avoidance mechanism. In NGN's environment, network carriers and service providers have to provide reliable and manageable services. Thus, SIP should adopt additional mechanisms to guarantee the reliability and fairness when UDP is used as the transport protocol.

SIP enables a wide set of applications ranging from Multimedia over IP to Instant Messaging, Presence, and Rich calls. Although the way to communication and the content may be quite different, the call setup procedures are almost the same for the reason that call control is separated from bearing in NGN. In this paper we focus on the calls, which may be either voice or video telephony. This study is applicable to other use cases where SIP is adapted for call control. The following sections are organized as the following. Section 2 introduces the related research on SIP's performance, and in Section 3 a transaction-based model is presented to estimate the call setup delay. In Section 4, a fine-granularity model is built to calculate the delay more accurately. Finally, simulation is taken to prove the effect of the models in Section 5.

2. Related Work

Lots of work has been done on the performance of SIP. In [5] the throughput variation with different versions of TCP (Reno, Vegas and Sack) used to transport SIP signaling traffic is described, and it draws the conclusion that TCP Vegas should not be recommended as a transport protocol for SIP signaling. In [6] the results of SIP call setup over 3G networks and other networks such as Intranet, LAN, WLAN are presented, and the comparison of those results is presented. However, all the data are got through measurement and no theoretic analysis is presented. In [7], different transport protocols that carry SIP are evaluated, and the results show that the fast retransmit algorithms and congestion control mechanisms make TCP and SCTP much better choices than UDP when carrying signaling traffic which is not very small.

None of the work mentioned above has presented a mechanism to analyze and calculate the delay in the phases of call setup, call duration or call release.

3. Modeling Performance of SIP

3.1 Performance Measurement Metrics

In this paper, we focus on the call setup delay, and so we only consider the metrics about delay. In ITU-T Recommendation E.721, call setup and release times are defined; and in G.114, guidelines on tolerable delay for a normal telephone conversation are provided. According to those, we define the phases of a SIP-based call setup process: Post-Dial Delay, Answer-Signal Delay, Re-negotiation Delay and Call-Release Delay. And here we only consider the first three delays and ignore the Call-Release Delay.

3.2 A Generalized Transaction-based Model

According to [9], how SIP operates over TCP and UDP is described. In [7], how SIP operated over SCTP is described. Here we summarize the way SIP operates over different transport protocols. SIP works as a signaling protocol, and thus much has to be done to guarantee the reliability.

For unreliable transport protocols, timeouts and application-layer retransmission implement the

hop-by-hop handshake. The client will not stop retransmitting the “INVITE” request until the “100 Trying” response is received. The server will retransmit the “100 Trying” each time it receives a retransmission of the “INVITE” request. “INVITE” requests are retransmitted first after 500 ms, with the interval doubled after that. Clients typically consider the next hop unreachable if no response has been received after the sixth retransmission. For reliable transports such as TCP and SCTP, the hop-by-hop handshake traverse a reliable transport connection, the client sends the “INVITE” over the transport connection and the server returns the “100 Trying” over the same connection. The transport layer undertakes the task of delivering the message to the next hop, so that no application-layer retransmissions are needed.

The summarization above shows that whether the transport is reliable or not, timeout-retransmission mechanism is taken on different layers to guarantee the reliability of SIP. Thus we can ignore the performance difference brought with implementing timeout-retransmission on different layers, and predict the delay of SIP-based call setup as the functions of loss rate, round-trip time and signaling messages’ size.

According to the specification on how SIP operates over TCP and SCTP in [9], SIP entities originate their connections from an ephemeral port, and mechanisms are provided to ensure that the response to a request and new requests sent in the forward direction can reuse the existing connection. However, new requests sent in the backward direction can’t reuse the existing connection, and they will originate new connections, on which responses to these backward requests are transported. This frequently causes a pair of SIP entities to use one connection for requests sent in each direction, which leads to potential scaling and performance problems. Some mechanisms can be utilized to solve the connection-reuse problem. In [10], a proposed mechanism is presented to reuse the existing connection by extending the “Via” header with a parameter named “Alias”. The parameter is utilized to indicate that the originator of the request wants to create a transport layer alias, that is, the “Alias” address becomes mapped to the actual IP address and port number observed as the source address of the current connection. In the following analysis, we assume that SIP works with the connection-reuse mechanism. Here we analyze the delay of SIP-based call setup when TCP is adopted as the transport protocol.

The total delay consists of four parts: post dial delay, answer signal delay, call release delay and call re-negotiation delay that is not marked. Associated with SDP, SIP fulfills the call setup function of call setup for voice, video and data applications, point-to-point and point-to-multipoint conferences. When calls are set up, RTP/RTCP establishes media channels and works for real-time media transfer. In the process of call, SIP’s capability of modifying sessions may be invoked to re-negotiate the communication. To best reflect SIP’s actual performance, the interval between “100 Trying” and “180 Ringing” is ignored, and the interval between “180 Ringing” and “200 OK” is assumed a constant that is deducted from the timeout value specified. Thus, the call setup delay consists of the following parts:

- D_T : Setup time for one connection setup (as far as TCP is concerned, it includes exchange of SYN, SYN-ACK, ACK messages; and as far as UDP is concerned, it is zero);
- D_S : Successfully transmitting all SIP and SDP messages, and it is where SIP works to set up the call;
- D_R : Successfully receiving an RTCP CNAME message.

The delay produced by RTCP CNAME message is located in the part of media transfer over RTP/RTCP. CNAME and BYE messages are more critical and needed to guarantee user perceived quality of service and the network performance. RTCP CNAME packet delay over wireless link is elaborately analyzed in [12].

The message flow between two User Agents is transaction-based, for example, (INVITE, 200OK, ACK) fulfills the call setup function and (BYE, 200OK) fulfills call release function. Here we present a model to estimate the delay of SIP-based call setup.

3.3 The INVITE Transaction Delay

Here we consider the case that UAC communicate with UAS without passing through any proxies. The INVITE transaction is similar to the three-way handshake in TCP's establishment. The process is described as the following stages. First, the UAC transmits its INVITE for $i \geq 0$ times unsuccessfully, and the $(i+1)$ -th INVITE arrives successfully at the UAS. The UAS will repeatedly retransmit its 200 OK until it receives an ACK. In general the UAS will send its 200 OK for $j \geq 0$ times unsuccessfully, and the $(j+1)$ -th 200 OK arrives successfully at the UAC. Similarly, the UAC will send ACK upon receiving the 200 OK. After $k \geq 0$ times unsuccessful transmission, the ACK arrives successfully at the UAS on the $(k+1)$ -th transmission. We consider the call is successfully set up on this point for the reason that the capability of one party who takes part in the communication is usually contained in ACK. Let $P_S(i, j, k)$ be the probability of having a three-way hand-shake consisting of exactly i failures transmitting INVITES, followed by one successful INVITE, followed by exactly j failures transmitting 200 OKs, followed by one successful 200 OK, followed by exactly k failures transmitting ACKs, and followed by a successful ACK. Thus

$$P_S(i, j, k) = p_f^i \cdot (1 - p_f) \cdot p_r^j \cdot (1 - p_r) \cdot p_f^k \cdot (1 - p_f) \quad (1)$$

The delay, $D_S(i, j, k)$, for this process, is expressed as

$$\begin{aligned} D_S(i, j, k) &= RTT + \left(\sum_{l=0}^{i-1} 2^l T_S\right) + \left(\sum_{l=0}^{j-1} 2^l T_S\right) + \left(\sum_{l=0}^{k-1} 2^l T_S\right) = RTT + (2^i - 1)T_S + (2^j - 1)T_S \\ &+ (2^k - 1)T_S = RTT + (2^i + 2^j + 2^k - 3)T_S \end{aligned} \quad (2)$$

The probability that D_S , the overall delay for a three-way handshake process, which is t seconds or less is expressed as

$$P[D_S \leq t] = \sum_{D_S(i, j, k) \leq t} P_S(i, j, k) \quad (3)$$

Then the expectation of D_S is expressed as

$$E(D_S) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} P_S(i, j, k) \cdot D_S(i, j, k)$$

Under the condition that loss rates are low enough the expected value of D_S is expressed as:

$$E[D_S] \approx \frac{3}{2} \cdot RTT + T_S \left(\frac{1-p_f}{1-2p_f} + \frac{1-p_r}{1-2p_r} + \frac{1-p_f}{1-2p_f} - 3 \right) \quad (4)$$

When p_r , p_f and p_f is low enough that most handshake is able to succeed before TCP abandons after its final trial. Among SIP transactions, only the one originated by INVITE consists of three messages, the others all consist of two messages, a request and its corresponding response (for example, INFO request and its corresponding response which may be used to carry the required media information), the above model can be simplified as the following (request is retransmitted $i \geq 0$ times and the response is retransmitted $j \geq 0$ times):

$$P_S(i, j) = p_f^i \cdot (1 - p_f) \cdot p_r^j \cdot (1 - p_r) \quad (5)$$

The call setup delay, $D_S(i, j)$, is expressed as:

$$D_S(i, j) = RTT + (2^i + 2^j - 2)T_S \quad (6)$$

The probability that the overall delay for the two-step transaction, is t seconds or less is the following:

$$P[D_S \leq t] = \sum_{D_S(i, j) \leq t} P_S(i, j)$$

Then the expectation of D_S is expressed as

$$E(D_S) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} P_S(i, j) \cdot D_S(i, j)$$

And the expected value of D_S is expressed as:

$$E[D_S] \approx RTT + T_S \left(\frac{1 - p_f}{1 - 2p_f} + \frac{1 - p_r}{1 - 2p_r} - 2 \right) \quad (7)$$

Judged from the solving process, it is shown that as for a transaction consisted of n steps, the total delay can be shown as the following expression:

$$E[D_S] \approx \frac{n}{2} \cdot RTT + T_S \left(\frac{1 - p_1}{1 - 2p_1} + \frac{1 - p_2}{1 - 2p_2} + \dots + \frac{1 - p_n}{1 - 2p_n} - n \right) \quad (8)$$

3.4 Analysis of RTCP CNAME Packet Delay

Here we define p (p_f or p_r) as the probability of a TCP segment in error, and $T \geq 5$ sec as the RTCP packet transmission interval. The end-to-end packet propagation delay is expressed with D (half of the RTT, usually 100 msec). For the reason that the CNAME is usually less than 500 bytes which is less than a TCP segment's length and the reason that we focus on the SIP's performance, we assume the CNAME takes only one TCP segment. Thus the RTCP CNAME Packet Delay is expressed as the following [12]:

$$D_R = RTT + \left(\frac{T}{2}\right)(1 - p) + \left(T + \frac{T}{2}\right)p(1 - p) + \left(2T + \frac{T}{2}\right)p^2(1 - p) + \dots = RTT + \left[T \frac{1 + p}{2(1 - p)}\right] \quad (9)$$

For the case of SIP over TCP, which is shown as Figure 1, before the peers interconnect with SIP, a TCP connection is established first. The three-way shake-hand process of TCP can also be analyzed with the above model for SIP's transaction. Although the transaction for TCP connection's establishment consists of three steps, the connection is considered as established once the SYN/ACK arrives at the active opener for the reason that in most application protocols, immediately after sending the ACK, the active opener sends a data segment to the passive opener that contains a redundant ACK which is the same as the above one. Thus the TCP connection setup delay can be analyzed with the equation for two-way transactions, that is, with equations (5)-(8). Thus, the total call setup delay (the length between ringing and the called peer's acknowledging the call setup is deducted) is expressed as the following:

$$D_{Total} = D_T + D_S + D_R \approx RTT + 2T_S \left(\frac{1 - p}{1 - 2p} - 1 \right) + \frac{3}{2} RTT + T_S \left(\frac{1 - p_f}{1 - 2p_f} + \frac{1 - p_r}{1 - 2p_r} + \frac{1 - p_f'}{1 - 2p_f'} - 3 \right) + RTT + T \cdot \frac{1 + p'}{2(1 - p')} = \frac{7}{2} RTT + T_S \left(2 \frac{1 - p}{1 - 2p} + \frac{1 - p_f}{1 - 2p_f} + \frac{1 - p_r}{1 - 2p_r} + \frac{1 - p_f'}{1 - 2p_f'} - 5 \right) + T \cdot \frac{1 + p'}{2(1 - p')} \quad (10)$$

The p , p_f , p_r , p_f' , p' is related with segment size, frame loss rate of their physical link. In the fol-

lowing part, they are calculated approximately according to their concrete link conditions.

4. Fine Granularity Performance Analysis

It's obvious that the transaction-based model is too coarse-granularity, and it ignores the concrete characteristics of transport protocols and thus may not be precise enough. Here we would analyze SIP's performance when TCP is adopted as its transport protocol.

Because the flows of SIP signaling for common call setup are so short that they often spend their entire lifetime in TCP's slow start phase, without suffering a single loss [14]. We make the following assumptions about the end-to-end TCP sessions carrying the SIP messages:

1. TCP operate in an interactive mode (this is valid since most of the transactions consist of a single request and its response).
2. The delayed acknowledgement mode of TCP operation is turned off, that is, the receiver sends an ACK when it receives every data segment.
3. The one-way delay for the message path is assumed to be 100msecs. The RTT is about 200 msec.
4. The TCP packet sizes for each SIP message are obtained from the actual SIP and SDP packets captured using a protocol analyzer.

Based on the above assumptions, a more detailed analysis on SIP over TCP is performed. The whole call setup consists of the following steps: TCP connection establishment, TCP's initial slow start and transferring the remainder after the first loss. The TCP connection establishment adopts the above transaction-based model all the same. Then the TCP's initial slow start is analyzed, with TCP's dynamic congestion window is taken into account.

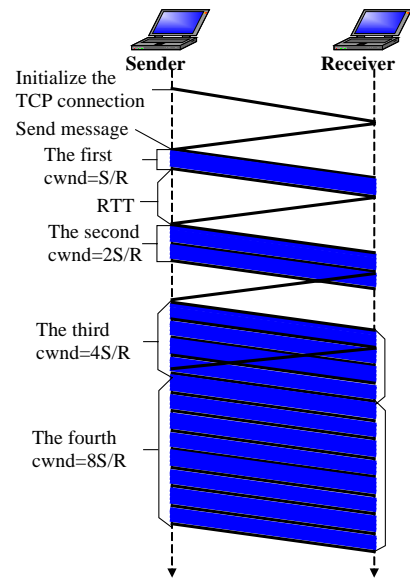


Fig. 1 TCP's slow start phase

4.1 SIP traffic in TCP's slow start

Here we analyze the SIP traffic over TCP during TCP's slow start if no congestion or loss occurs. The sender initiates with the congestion window $cwnd=1$ segment, and sends the first segment to the receiver. Whenever the sender receives an ACK segment, it increases a $cwnd$, and thus the $cwnd$ doubles after one RTT analogically if no congestion or loss occurs. Here we consider the case that communication is between the two User Agents only, shown as Fig.1. Let O be the total length of the SIP traffic during the slow start phase over TCP before the first loss, and S be the length for one segment. And the total number of segments of the SIP traffic is O/S . The general form of the number of segments in the k -th $cwnd$ is 2^{k-1} . Let K be the number of $cwnd$. Thus,

$$K = \min\{k : 2^0 + 2^1 + \dots + 2^{k-1} \geq O/S\} = \min\{k : 2^k - 1 \geq O/S\} = \min\{k : k \geq \log_2(1 + O/S)\} \\ = \lceil \log_2(1 + O/S) \rceil \quad (11)$$

According to Fig.1, the sender may be in idle state after sending the total data of the $cwnd$, waiting for the ACK segment. But after sending the whole data of the third $cwnd$, the sender can proceed with the

throughput model presented in [16]. Where

$$E[T_{remain}] = E[d_{remain}] / R(p, RTT, T_0, W_{max}) \quad (22)$$

$$R = \begin{cases} \frac{\frac{1-p}{p} + \frac{W(p)}{2} + Q(p, W(p))}{RTT(\frac{b}{2}W(p)+1) + \frac{Q(p, W(p))G(p)T_0}{1-p}}, & \text{if } W(p) < W_{max} \\ \frac{\frac{1-p}{p} + \frac{W_{max}}{2} + Q(p, W_{max})}{RTT(\frac{b}{8}W_{max} + \frac{1-p}{pW_{max}} + 2) + \frac{Q(p, max)G(p)T_0}{1-p}}, & \text{otherwise} \end{cases} \quad (23)$$

$$W(p) = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \quad (24)$$

And T_0 is the average retransmission delay, W_{max} is the maximum window constraint. Synthetically, with the fine model, the total delay for transferring the SIP traffic is expressed as the following:

$$D_{S_Total} = D_{SlowStart} + T_{loss} + T_{remainder} \quad (25)$$

Thus, we synthesize the delay, and revise the expression (10) to the following modified expression:

$$D_{Total} = D_T + D_{S_Total} + D_R = RTT + 2T_S \left(\frac{1-p}{1-2p} - 1 \right) + RTT + \frac{O}{R} + P \left(RTT + \frac{S}{R} \right) - (2^p - 1) \frac{S}{R} + p_{loss} \cdot \left(Q(p, \frac{O+1}{2}) \cdot E[Z^{T_0}] + (1 - Q(p, \frac{O+1}{2})) \cdot RTT \right) + \frac{d-O}{R(p, RTT, T_0, W_{max})} + RTT + T \cdot \frac{1+p}{2(1-p)} \quad (26)$$

5. Simulations and Evaluation of Models

We take measures to verify the models in two steps, the first step is to prove that the transaction-based model can forecast the call setup delay with a tolerable error, and the second step is to compare the results from transaction-based model, fine-granularity model and measuring.

TABLE 1 Message Sizes with SIP (SDP Loaded) Call Setup Procedure

Messages	INVITE	200 OK	ACK	RTCP CNAME
Payload Size	586 octets	340 octets	326 octets	120 octets

Step 1: Here we adopt the average RTT (which is 220 msec) got from the measurements with ethereal in a LAN environment, T_S is 500 msec, and T is 5s. p_r , p_f and p'_f are calculated with their concrete link conditions and sizes. It's obvious that the function in (10) is incremental monotonously when p (which the general form of p_r , p_f and p'_f) increases in the area (0,1). And we get the results of various links from simulations and the transaction-based model, the comparison is shown in Fig.2. We can find that the average result got from simulations is always higher than that got from the transaction-based model, and the reason is that in the transaction-based model we neglect the slow start characteristic and the delay is estimated lower.

Step 2: Here we compare the results got from fine-granularity model, transaction-based model and simulations. Within the fine-granularity model, the INVITE and ACK requests (whose sum is 912 octets) are sent in one direction and the 200 OK is sent in another direction (which follows 100 trying and 180 ringing messages actually). We can find that the result got from the fine-granularity is higher than that got

Schooler, SIP: The Session Initiation Protocol, *IETF RFC 3261*, 2002

[10] R. Mahy, "Connection Reuse in the Session Initiation Protocol (SIP)," *IETF Internet-Draft-ietf-sip-connection-reuse-02*, 2004

[11] ITU-T, Recommendation E.721, *Network Grade of Service Parameters and Target Values for Circuit-Switched Services in the Evolving ISDN*, May. 1999.

[12] S.K. Das, E. Lee, K. Basu, S.K. Sen, "Performance optimization of VoIP calls over wireless links using H.323 protocol," *IEEE Transactions on Computers*, Volume: 52, Issue: 6, Pages: 742 -752, Jun. 2003

[13] J. Rosenberg, H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)," RFC3264, IETF, 2002

[14] N. Cardwell, S. Savage, T. Anderson, "Modeling TCP Latency," *Proceedings of the 2000 IEEE INFOCOM Conference*, Pages: 1742-1751, Mar. 2000

[15] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM '98*, Sep. 1998

[16] J. Padhye, V. Firoiu, D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Technical Report 99-02, University of Massachusetts, 1999