Universität Stuttgart

INSTITUT FÜR
KOMMUNIKATIONSNETZE
UND RECHNERSYSTEME
Prof. Dr.-Ing. Dr. h. c. mult. P. J. Kühn

Copyright Notice

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: +49-711-685-68026, Fax: +49-711-685-67983
email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Measurement of the SIP Parsing Performance in the SIP Express Router

Stephan Wanke[1], Michael Scharf[1], Sebastian Kiesel[1], and Stefan Wahl[2]

[1] Institute of Communication Networks and Computer Engineering (IKR), University of Stuttgart, Germany
[2] Alcatel-Lucent Deutschland AG, Research & Innovation Germany

Abstract. Future telephony and multimedia systems will use the Session Initiation Protocol (SIP) for signaling purposes. SIP is a text-based protocol that imposes challenges for an efficient message processing. The ability of SIP entities to process SIP messages quickly is crucial for the performance of these networks, which often have strict timing requirements, e. g., to keep the call setup delays small.
This paper studies the performance of SIP message processing in SIP proxies, focusing mainly on the impact of message parsing. We perform a detailed delay analysis for the widely used SIP Express Router (SER). Our measurements show that message parsing actually contributes significantly to a SIP proxy's processing efforts, and therefore confirm other existing studies. However, our results also show that the overall delay in high-performance SIP proxies is stronger affected by other factors, in particular the operating system.

1 Introduction

The Session Initiation Protocol (SIP) is the de facto standard for session signaling in IP-based telephony and multimedia systems, such as the 3GPP IP Multimedia Subsystem (IMS) [1] or ETSI TISPAN. In these systems, many entities are involved in call control and many SIP messages are exchanged for time-critical signaling flows like call setups. Signaling latencies depend on a variety of factors, including both transport and processing delays. This is why the ability of SIP entities to process SIP messages effectively is crucial for the overall network performance. SIP is a text-based protocol. Therefore, SIP entities have to parse text messages, which is reported in literature to consume significant amounts of processing time [2]. This has also fostered proposals to use alternative encoding formats for session signaling; a recent example is, e. g., [3].

The purpose of this paper is to understand the performance impact of SIP message parsing. In spite of many research efforts on SIP signaling, surprisingly little work has been published on the performance of SIP entities. We report detailed performance measurements that have been obtained from the SIP Express Router, which is a widely used high-performance SIP proxy. They confirm that SIP parsing indeed consumes significant processing power. However, our results

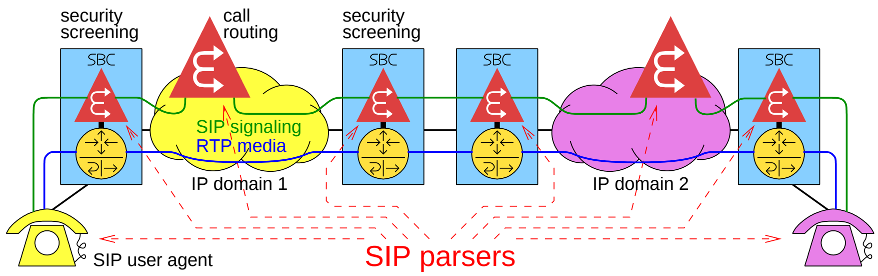

Fig. 1. SIP protocol entities in a carrier-grade VoIP network (simplified)

also reveal that other factors such as the operating system have a very significant impact, too. These effects are hardly addressed in other published studies.

The remainder of this paper is structured as follows: Section 2 gives an overview of SIP performance in signaling systems and also reviews related work. In Section 3, the testbed for our measurements is presented. In Section 4, we report the results of our measurements. Finally, Section 5 concludes this paper.

2 SIP Performance in Signaling Systems

Carrier-grade Voice-over-IP (VoIP) or multimedia networks as the one sketched in Fig. 1 include several entities that process SIP messages: For instance, calls are handled by entities realizing Call Session Control Functions (CSCF), and security screening at domain boundaries may be realized by Session Border Controllers (SBC). Even for simple actions several SIP messages have to be exchanged and processed by several entities. As a consequence, it is important to quantify the delays resulting from SIP message processing in the different entities.

2.1 SIP Proxies

In addition to the user agents in the end systems, SIP messages are processed by SIP proxies, SIP Back-to-Back User Agents (B2BUA), or network elements such as SBCs that include a proxy or B2BUA. There are different SIP proxy implementations, both commercial ones and open-source solutions. In this study, we selected the SIP Express Router (SER) [4], since it is an open source proxy that is widely deployed and known for its efficiency and high scalability.

As illustrated in Fig. 2, SER is divided into a core, which implements the basic functionalities, and so-called extensions modules. The configuration of SER is performed by a file with a shell-like syntax. This file provides so-called routes that determine the handling of received SIP messages. Additional modules extend the core functionalities, e. g. for registration. SER is completely implemented in C. The SER SIP parser is optimized for performance and uses pre-calculated 32-bit hash tables for effective parsing [5]. Because of its high-performance design SER does not parse whole SIP messages, but only the relevant parts, depending on the configuration.

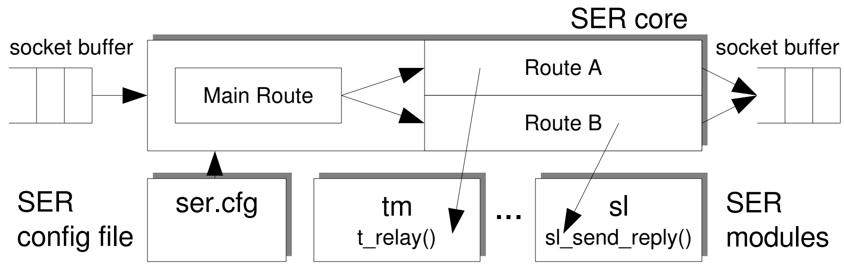

Fig. 2. Structure of the SIP Express Router

2.2 Related Work

Despite the fact that performance is important for the deployment of SIP in carrier platforms, there is only few published work that analyzes the performance of SIP proxies in depth, in particular concerning the impact of different processing steps. The performance of SIP entities largely depends on the implementation. Still, performance studies and capacity planning require at least some estimates.

A detailed study on SIP performance was published in [2]. The paper compares different proxy implementations that were specifically programmed for this study. It is reported there that the parsing needs about 25 % of the processing time, and that the total processing time of a single SIP message ranges between 1.8 and 0.2 ms. In [6] the performance of the JAIN SIP stack has been measured and call setup delays have been determined. The latency of J2EE SIP application servers is studied in [7]. However, these measurements only consider the proxy as black box, and the effort of different SIP processing functions has not been considered in detail. Other simulation studies either take the service time in proxies just as a variable (e. g., [8]), or simply make some assumptions [9].

There are also discussions on using alternative encodings for session signaling, such as type-length-value or XML. A recent example is [3]. It mainly depends on the parsing whether such formats improve the performance compared to the UTF-8 encoded text format of SIP. In the rest of this paper, the SER parsing is analyzed, which quantifies the maximum potential saving of such alternatives.

3 Measurement Setup

3.1 Testbed

For our measurements we developed a testbed that allows to measure the performance of SER by using different tools under varying parameters. Its structure is shown in Fig. 3. We use two SIPp load generators [10] to emulate the User Agents and to generate the SIP messages. We configured the two load generators so that they exchange the SIP message sequence shown in Fig. 4. They communicate over the SER SIP proxy with each other. As transport protocol for

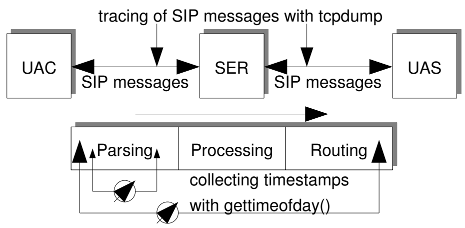

Fig. 3. Structure of the test bed

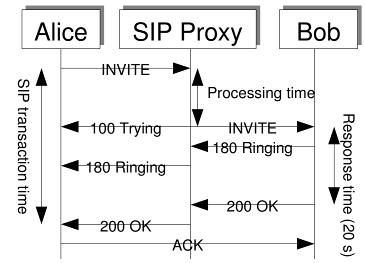

Fig. 4. Signaling flow for a SIP transaction to establish a session between two entities

SIP we use UDP. The configuration of the SER for the measurements was rudimentary, for example, without Authentication, Authorization and Accounting (AAA) functions. This allows to isolate the effects from the SIP processing of SER, as it prevents the influence of auxiliary protocols like RADIUS and DNS. The computers with SER and the load generators are connected via 100 Mbit/s FastEthernet interfaces. We use a Pentium IV 2.8 GHz with 2 GB RAM and a SUN Solaris Workstation with two Ultrasparc III+ 1 GHz processors with 2 GB RAM for comparison. It must be noted that under Solaris we use the gcc compiler for compilation of the SER, not an optimizing SUN C compiler.

3.2 Measurement Methodology

There are many different metrics to quantify SIP performance [11]. In this paper, we focus on the delay of a single SIP proxy. We study the sojourn time, i. e., the processing delay in the proxy (see Fig. 3). This is a critical metric because the summation of processing delays contributes to the total signaling delay.

We use three different methods to analyze the performance of the SER: First, we use tcpdump to capture Ethernet frames. In order to quantify the time the SIP message spent in the computer hosting SER, we calculate the difference of the time stamps from Ethernet frames carrying SIP INVITE messages from and to SER. Second, we inserted "gettimeofday()" statements in the source code of the SER to measure the time that is needed for interesting code sections, in particular around the parsing and the total processing code section. Finally we use a profiler to collect data at run-time. From the network's point of view, the data collected with tcpdump is the most important one because this is the delay SIP messages experience traveling through the proxy.

Our measurements include certain systematic errors: When the code is extended for measurements, these extensions have to be executed, which costs time. Moreover, the accuracy of the "gettimeofday()" function is limited. The accuracy of tcpdump time stamps is limited, too, and there are also small errors due to media access. The results of the run-time profiler analysis are only relative performance values because the profiler extends the machine code extensively.

4 Measurement Results

With the testbed presented in the last section we performed measurements to investigate major influence factors on SIP proxy performance. We could identify three main factors for the performance of the SER: The format of the used SIP messages, the operating system, and the configuration of SER.

4.1 Internal and External Processing Times

The external processing times are derived from time stamps of Ethernet frames. They differ from the internal processing time determined with the "gettimeofday()" function. This is because the host's CPU is not only needed by the SER process, but also by the operating system for handling Ethernet frames and for executing other processes.

In Fig. 5, the external and internal processing times and the parse time are shown as functions of the rate of SIP INVITE messages. As to be expected, the parsing time determined by the "gettimeofday()" functions is load independent. The internal processing time is slightly load dependent. A possible reason for this is the interruption of the SER process by the Network Interface Card (NIC) when receiving Ethernet frames. However, the external processing time is highly load dependent. It grows from 200 μs to over 4 ms for 2000 Call Attempts Per Second (CAPS). The share of time needed for parsing related to the internal processing time is in the range of 20 to 33 %.

The results of the profiler analysis is illustrated in Tab. 1. Our measurement confirm the results of [2]: 25 % of the time is needed for parsing. The main share of 50 % is needed for the stateful forwarding of SIP request and 15 % are needed for the stateless forwarding of SIP replies. Note that the result of the profiler analysis is a superposition of SIP requests (INVITE messages; transaction stateful processing) and replies (100 Trying, 200 OK, ...; stateless processing). In contrast, the data collected with the other methods focuses on requests (INVITE messages) only.

The difference between internal and external processing time is the time spent in the system outside the SER process (e. g., operating system kernel and NIC drivers). For high rates of SIP INVITE message, the share of time needed for processing of messages in the operating system exceeds the time needed for parsing and processing of the messages inside the SER process by orders of magnitude. It can be assumed that the process scheduler of the operating system is accountable for the increase of the external processing time for higher loads, because it interrupts the processing of SIP messages through SER. However, varying the Linux kernel scheduler frequency (100 Hz, 250 Hz and 1000 Hz) did not show any significant impact.

4.2 Influence of the Format of SIP Messages

In order to study the impact of differently formatted SIP messages, we performed measurements with different SIP messages. We focus on RFC 3261 compliant messages and do not examine the influence of corrupted messages.

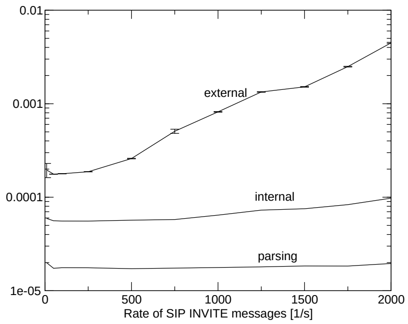

Fig. 5. External, internal processing time and parsing time (in s) for Linux

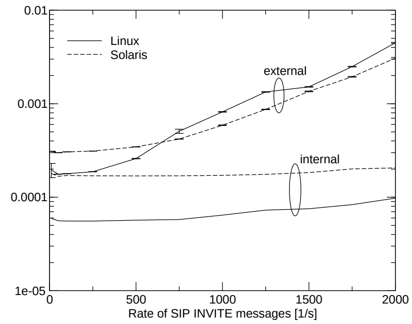

Fig. 6. Comparison of the processing times (in s) between Linux and Solaris

In Tab. 2, the parsing times for differently formatted messages are presented: Message 1 is a standard formatted SIP message. Message 2 has its via header at the end of the message. In message 3, additional header fields are inserted between the via header and the rest of the SIP message. Additional to message 3, in message 4 multi-line header fields and combination of upper case and lower case characters in the header names are used. Combination of upper and lower cases in the header names increase the number of comparisons to determine the type of the header, as multi-line header fields do.

The SER has to parse all the via headers to add its own via header at the end. One can see that the format of SIP messages has a significant influence of the parsing time. Sloppily formatted SIP messages can increase the parsing time. However, the parsing time is still clearly one or two orders of magnitude smaller than the external processing time.

4.3 Impact of the Operating System

Furthermore, we analyze the influence of the operating system in more detail. We compare Linux (kernel version 2.6.18) with Solaris (version 5.10) both with default configuration except for optimized UDP settings.

Table 1. Relative time shares of the processing stages collected with the profiler

Action	Share [%]
Parsing	25
Stateful processing	50
Stateless processing	15
Others	10

Table 2. Parsing time for different formatted, RFC 3261 compliant SIP messages

Load	Parse time [μs]
Message type 1	18.0
Message type 2	12.5
Message type 3	20.0
Message type 4	22.5

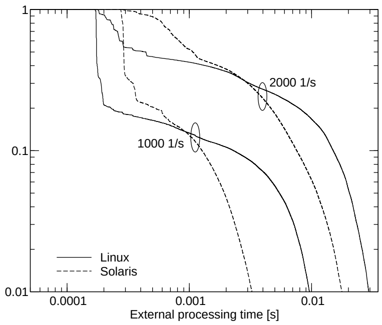

Fig. 7. CCDF of the external processing time for Linux and Solaris for 1000 and 2000 CAPS

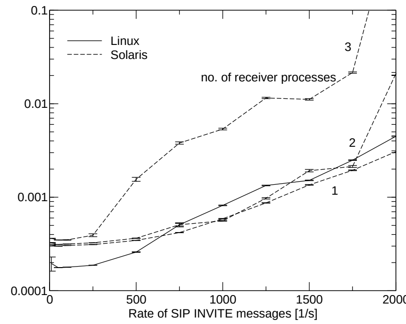

Fig. 8. External processing time (in s) for Linux and Solaris with different number of receiver processes

In Fig. 6, the external and internal processing times are shown as functions of the rate of SIP INVITE messages, for Linux and Solaris. The absolute values are not easy to compare because the hard- and software architectures of Linux/x86 and Solaris/Ultrasparc are quite different. Still, for both operating systems, the same effects can be observed: The external processing time increases with increasing of the SIP message load. It is interesting to note that for higher loads Solaris performs better in the external processing time, although the internal processing time of the Solaris system is higher than when using Linux.

So far, we have only discussed mean values of the processing time. In Fig. 7 we present the Complementary Cumulative Distribution Function (CCDF) of the external processing time for Linux and Solaris, obtained from the tcpdump time stamps. The Linux and Solaris curves are similar. For increasing load the jitter increases but the minimum external processing time keeps constant. For Linux the 95 % quantile increases from 5 ms to over 20 ms and for Solaris it increases from 2 ms to 11 ms, when the CAPS are increased from 1000 to 2000. In both systems there is a non-negligible probability of rather high delays.

4.4 Impact of the SER Configuration

In Fig. 8 the external processing time is shown as a function of the rate of SIP INVITE messages. As parameter we change the number of receiver processes. They can be adjusted by the SER configuration file. When running on Linux, the number of receiver processes has no influence on the processing time. This is why in Fig. 8 only the result for one receiver process is illustrated. However, if Solaris is used, the number of receiver processes has an impact on the processing time: It can be observed that there is a significant increase of the external processing time when the number of receiver processes exceeds the number of CPUs.

5 Conclusion and Future Work

In this paper we present detailed performance measurements for the SIP Express Router (SER). We determine three major factors influencing the performance of SER: The format of SIP messages, the operating system, and the configuration of SER. Our results confirm previous studies, which have reported that the parsing may take up to 25 % of the internal processing effort. However, we also show that the parsing effort has a much smaller share of the external delay, which is the critical performance metric from the network's point of view. Even though this paper only considers the SER proxy, we expect that other SIP entity implementations have similar performance characteristics.

The measurements indicate that the potential for speeding up session setup by using alternative message encoding formats is rather low, in particular when there are also further delays caused by, for instance, address resolution or AAA functions. Still, since the format of SIP messages does affect the processing complexity, it could be useful to normalize SIP messages at domain boundaries, such as to re-arrange headers in order to reduce processing efforts in core SIP proxies.

Our measurements results have been obtained from a simple network setup. A further step would be to perform measurements in a complete signaling scenario with AAA functions and other auxiliary protocols, e. g., in a full IMS setup.

References

1. Camarillo, G., García-Martín, M.A.: The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds. 2 edn. Wiley (2005)
2. Cortez, M., Ensor, J.R., Esteban, J.O.: On SIP Performance. Bell Labs Technical Journal **9**(3) (2004) 155–172
3. Baset, S., Schulzrinne, H.: Peer-to-Peer Protocol (P2PP). IETF Internet Draft, work in progress, (February 2007)
4. SIP Express Router (SER). http://www.iptel.org/ser/
5. Janak, J., Kuthan, J.: SIP Express Router v0.8.8 - Developer's Guide. FhG Fokus (2002)
6. Gokhale, S.S., Lu, J.: Signaling Performance of SIP Based VoIP: A Measurment-Based Approach. In: Proc. IEEE Globecom. (November 2005)
7. Van Den Bossche, B., De Turck, F., Dhoedt, B., Demeester, P., Maas, G., Moreels, J., Van Vlerken, B., Pollet, T.: J2EE-based Middleware for Low Latency Service Enabling Platforms. In: Proc. IEEE Globecom. (November 2006)
8. Gurbani, V.K., Jagadeesan, L.J., Mendiratta, V.B.: Characterizing Session Initiation Protocol (SIP) Network Performance and Reliability. In: Proc. ISAS 2005, LNCS 3694. (2005) 196–211
9. Kist, A.A., Harris, R.J.: SIP Signalling Delay in 3GPP. In: Proc. IFIP International Symposium on Communication Interworking. (2002)
10. Richard Gayraud, Olivier Jaques et al.: SIPp - SIP Load Generator. http://sipp.sourceforge.net/index.html
11. Malas, D.: SIP End-to-End Performance Metrics. IETF Internet Draft, work in progress (January 2007)