SIP Registration Stress Test

Miroslav Voznak and Jan Rozhon
Department of Telecommunications
VSB – Technical University of Ostrava
17. listopadu 15/2172, 708 33 Ostrava Poruba
CZECH REPUBLIC
miroslav.voznak@vsb.cz, jan.rozhon@gmail.com

Abstract: - This paper deals with a benchmarking of SIP infrastructure and improves the methodology of SIP performance evaluation further to better fit into the design of the SIP testing platform, which is being designed in the VSB – Technical University of Ostrava. By separating registrations from calls, we were able to measure both cases without the need of extensive postprocessing of data to ensure the data in one case is not affected by the ones from the other case. Moreover the security vulnerability of the SIP protocol has been harnessed to allow measuring software for performing both registrations and calls together but in individual processes, which builds the basis for planned and already mentioned modular design of the platform. In this paper, we present the results from separate registration stress tests and we explain the usage of the proposed SIP benchmarking methodology.

Key-words: SIP, Registration,Benchmarking, RFC 6076, Stress test, SIPp.

1 Introduction

With our methodology for testing and benchmarking SIP infrastructure finished, we had the opportunity to perform several series of tests on multiple different platforms. From these tests we realized, that it would be very beneficial to modify the existing testing platform to allow us for performing separate test scenarios on each of the important SIP dialogs. This way the movement towards the modular design started. During this work at the beginning of this year the new RFC 6076 was adopted finally standardizing most essential measured parameters [1].

With the parameters standardized we have developed the most important testing scenarios – the registration test scenario and the call test scenario, both having its roots in the previously used scenario for complex performance measuring. Each of those scenarios offers a different perspective when defining the SIP server limits and can be run either separately to test some special environments or occasions or simultaneously to simulate the real VoIP client behavior. The latter presented a big challenge, because the testing software does not allow running multiple scenarios at once inherently. However this problem was walked around by exploiting SIP security vulnerability, which allows a client from one address register another. This way the basis of module based testing platform has been created.

In this paper we present the example of results gained by testing two different versions of most commonly used VoIP PBX Asterisk focusing on its ability to handle multiple simultaneous registrations coming in several consequent bursts [2]. This example is particularly useful to determine how the SIP server reacts in the case of network failure and consequent restoration of full connectivity, when all the clients try to register at once.

In the given examples the way how the SIP server responds to bursts with high loads can be determined and all the conclusions are made according to information obtained by the measurements on the client side exclusively, because the measurements on the server side are often impossible due to the provider restrictions.

2 State of the Art

As stated in the introduction the authors have vast knowledge in the field of performance measurement of SIP servers using open source software testing tool SIPp and cooperatively created the basic methodology and testing platform. This methodology had its foundations in the RFC-draft, which was adopted by the IETF as RFC 6076 this year; therefore the existing methodology is almost entirely compliant with it [1].

The formerly used testing platform utilized one complex SIP scenario to test both registrations and calls at once, which allowed for complex analysis of

ISBN: 978-1-61804-077-0

the SIP server, but inherently resulted in the call measurement to be affected by the registrations [3], [4]. This issue could have been solved by data postprocessing, when the number of actually created calls was taken as the basis instead of desired call load, but except of this the user of the testing platform could not have simply chosen what type of test he wants and moreover the more complex scenarios which would include automatic answers and more sophisticated SIP dialogs could not have been created. For this reason the modular approach has been adopted [5], [6].

Apart of the mentioned the proprietary solutions also exist, but they offer limited or nonexistent compliance with IETF standards and could not be considered cost effective [1], [7], [8].

3 Testing Platform and Scenario

For the registration test the testing platform must slightly differ from the one presented in complex methodology. The main reason for this comes from the lack of need for UAS part of the testing platform, since only end to end dialogs between client and SIP server will occur. Basically all the computers will act as the initiators of the SIP registration dialogs, which is why they are going to be called UACs (User Agent Servers). For the generation of SIP messages, the well-known testing tool SIPp will be used and to ensure that client computers will not run out of hardware resources, the generated load will be spread among 8 computers. From these assumptions the basic test topology will look as depicted on the Fig. 1.

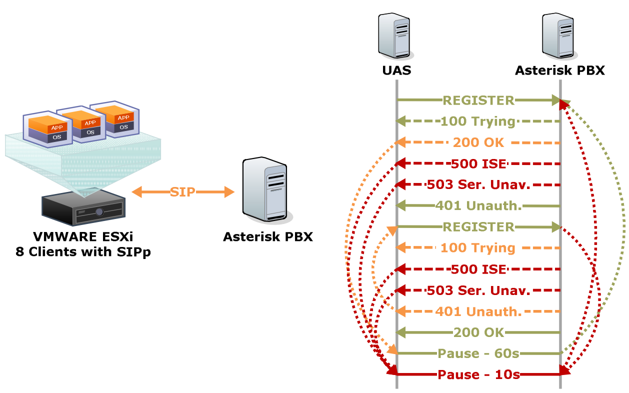

Figure 1. Test topology and flow of SIP messages during the registration dialog.

Correct messages are colored in green, while unexpected messages are orange and error messages red. Dotted lines with arrows represent the hop after error or unexpected message is received, or error occurs while transmitting the message.

The virtualization is a useful option because due to the internal limitations of SIPp testing tool it is better to distribute individual SIPp instances each on the separate computer, therefore the usage of physical computers would result in large space and networking requirements.

The main configuration elements of the virtualized computers can be summarized in these points:

- Guest OS Ubuntu 10.10 x64 Server Edition 1x Virtual x64 Processor Core @ 2.8 GHz
- 2048 MB Virtual RAM
- UDP Buffer Size 131 071 B (default size)
- File Descriptor Limit 65535 (ulimit -n)
- Stack Size Limit unlimited (ulimit -s)

The keystone element of the platform – the SIP server is realized on the separate hardware machine with this configuration:

- OS Ubuntu 10.04 x64 Server Edition
- AMD Athlon X2
- 4 GB DDR2 RAM
- UDP Buffer Size 131 071 B (default size)
- File Descriptor Limit 100000 (ulimit -n)
- Stack Size Limit unlimited (ulimit -s)
- Asterisk 1.6.2.16.2
- Asterisk 1.8.2.4
- 100 000 SIP Peers

Both network elements – the host virtualization server and SIP server are interconnected via a gigabit switch to ensure minimal additional delays caused by the network infrastructure.

The measurement is performed only on client devices to reflect the practical situation, when SIP server is not accessible to perform measurement of any kind. On the client devices these values are measured:

- Number of individual SIP messages
- Number of retransmissions
- Number of Timeouts
- Approximated RRD

All the mentioned parameters will be used to determine the SIP servers performance and the last one will be properly described and explained in the next section.

On the Fig. 1 the message flow of the registration dialog is also depicted. The standard messages of the successful registration dialog are colored in green and are followed by 60s long pause after which the reregistration takes place. Additionally out of sequence messages can be received from the SIP server, when the load exceeds the level SIP server can handle without significant delays. These

ISBN: 978-1-61804-077-0

messages are valid part of the SIP dialog and are colored in orange. Each time such a message is received the jump to correct part of the dialog is performed. When the load is even higher, the error messages or timeouts might occur. In this case two 5XX messages are being anticipated and when one of these messages is received or one of the correct dialog messages times out the error jump is performed to a 10s long pause after which another attempt to register is sent.

4 Methodology and Limit Definition

The aim of the measurements is to determine how the SIP server Asterisk will respond to high burst loads of SIP registrations. These registrations will be sent in 5 consecutive 1 second long bursts with a given load and the client devices will try to hold these registrations for 15 minutes by sending reregistration requests every 60 second. After the 15 minutes all the measured parameters are logged and the process repeats again with the higher load. If the registration attempt is not successful no matter if this is caused by an error message or timeout, the client will wait for 10 seconds before it tries to register again. This way the SIP server is given the possibility to spread the load to longer period and the real world VoIP client behavior is preserved. Although the way the error messages and timeouts are treated is the same, the severity of these two kinds of errors is different. While receiving the error message causes client not to be able to register for about 10 seconds (the error pause interval), after the message timeout this period is about 42 seconds, which is caused by the timeout mechanism of the SIP protocol. For this reason the timeouts have greater impact on the SIP server's performance evaluation.

Due to the length of the test the client will attempt to send Register message 15 times. In this number the retransmissions are not counted. From this number the limit for determining whether SIP server passed the test successfully or not can be derived. If the number of timeouts exceeds the 1/15 of the total number of unique Register requests sent, it can be interpreted as the clients were not able to register successfully after 45 seconds. To ensure that SIP server has the possibility to recover from the burst, this limit was doubled. Same calculation can be made for the error messages, but with the lower weight caused by the significantly shorter period when the client cannot register. Because no 5XX error message was received during the whole test, this paper's result analysis will work only with the number of 2/15 (~13%) timeouts as limit for successful passing the test.

As defined in the previous section, the approximated RRD is also measured. In the previous work and in the RFC 6076, the RRD (Registration Request Delay) is defined as the time between sending the Register request and receiving the 200 OK response. Approximated RRD in this paper is measured exactly the same way, but due to the limitations of the SIPp in loop scenarios, the time measurement is not possible by the available timers and must be performed by the logging mechanism of the SIPp. To the log no precise time can be written. The most useful possibility is to use the internal clock ticks of the SIPp. One clock tick is loosely comparable with the 1 millisecond, but the precision may vary in higher loads, therefore the measured parameter can be viewed only as the fair approximation of the RRD. Approximated RRD is therefore important not for its absolute values but for its trend change while the load is being increased.

5 Result Analysis

In this section the results will be reviewed. In two subsections four charts will be presented and on each of these charts the Asterisk 1.6 will be displayed by the dark brown rounded symbols, while Asterisk 1.8 will be depicted by orange triangles. All the measured parameters are related to the number of simultaneous registrations and each dot in the chart represent a single 15 minutes long step in the test process.

5.1 Successful Registration Attempts and SIP Timeouts

Successful Registration Attempts display the ratio between attempted registrations and success indicating 200 OK responses. The best and optimal value is 100 %, but no architecture is designed to handle huge number off registrations at once, therefore mechanisms which will spread the load to longer interval if the unsuccessful registration occurs are implemented. As mentioned in the previous section the threshold coming from number of timeouts was designated to 13% and because the timeouts were the only error measured during the whole test, this threshold can be used directly on the number of Successful Registration Attempts.

The charts on the Fig. 2 show clearly the difference between two tested architectures. While Asterisk 1.6 starts having problems when the load of 1000 registrations is generated and falls under the designated threshold immediately after 1280 simultaneous registrations, Asterisk 1.8 holds the 100% ratio for the whole time of the test reaching

ISBN: 978-1-61804-077-0

stable behavior even with the load of 6400 simultaneous registrations.

The similar behavior can be seen on the chart depicting the number of SIP timeouts. For Asterisk 1.6 timeouts are the essential problem, on the other hand no timeout occurred while testing Asterisk 1.8. From this information can be stated that new version of Asterisk handles the burst load of SIP registrations much better than the older one. On this place it would be good to state, that all the parameters were set to defaults on both Asterisks, and same number of SIP peers was used, therefore no external event could have influenced the results.

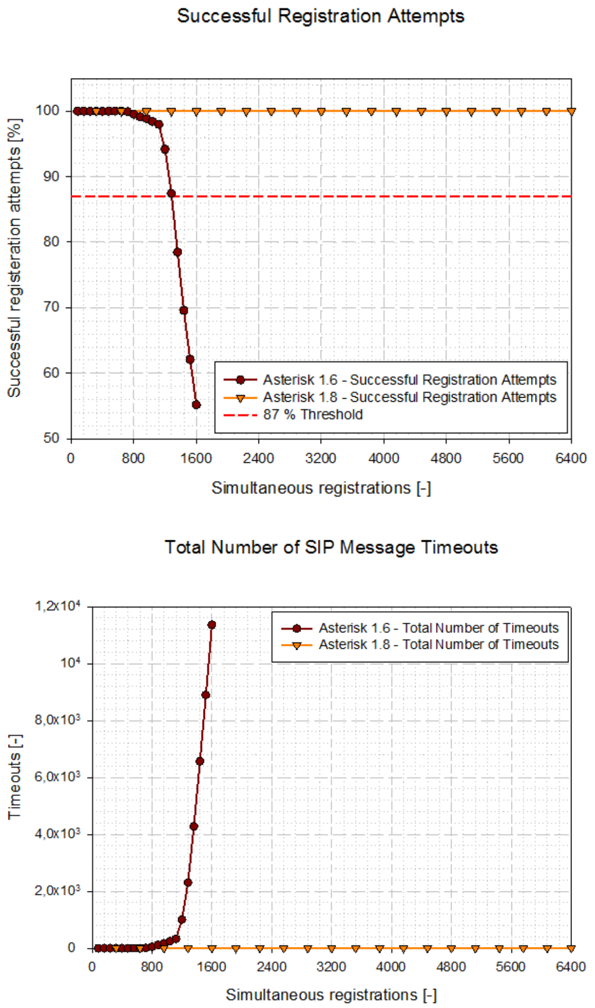

Figure 2. Successful Registration Attempts and Total Number of SIP Message Timeouts.

5.2 Retransmissions and Approximated RRD

From already presented information the clear assumption about what Asterisk is better in case of network connectivity failure and subsequent recovery can be made. Now we are going to explore whether the retransmissions and approximated RRD will confirm this assumption.

Approximated RRD was clearly explained in the previous section, therefore no further explanation will be presented. The retransmissions occur when there is long period between sending the message and receiving the appropriate answer. In SIP the standard timer define, that the first retransmission takes place when the response is not received in 500 milliseconds after sending the request. Each subsequent retransmission is sent after doubled interval, until the 4 seconds are reached. When this happens all other retransmission will be sent after 4 seconds giving the following sequence of time periods between the neighboring messages – 500ms, 1s, 2s, 4s, 4s… After the sequence of 9 retransmissions the timeout occurs. The number of retransmissions increases when SIP server cannot successfully process all the messages. In the following charts, the total number of retransmissions means the sum of all retransmissions of all messages in the SIP registration dialog.

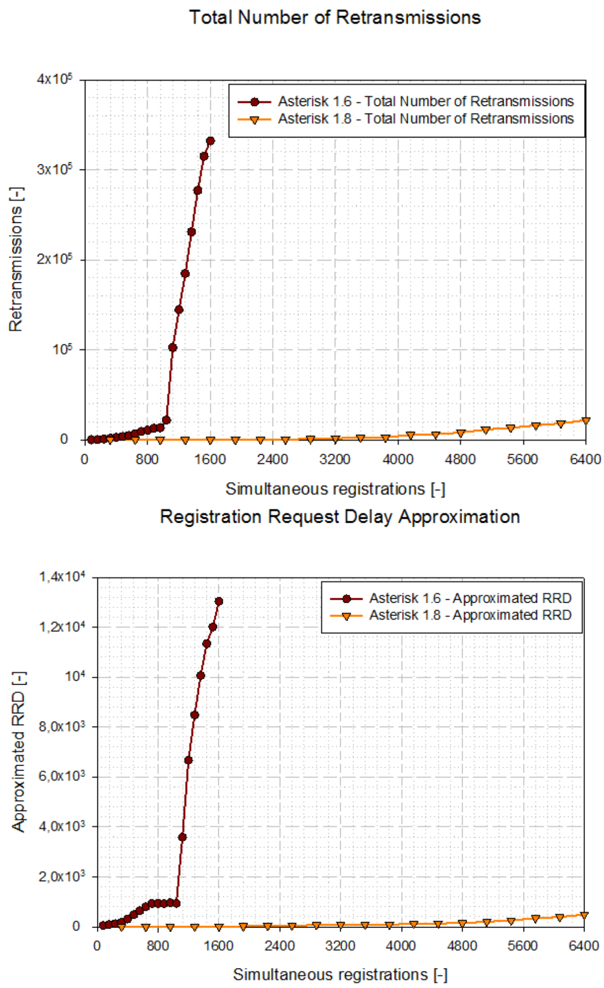

Figure 3. Total Number of Retransmissions and Approximated Registration Request Delay.

Both parameters depicted on the Fig. 3 – Total Number of Retransmissions and approximated RRD – share the same trend and underline the knowledge obtained in the previous subsection. Asterisk 1.8 excels again reaching minimal number of retransmissions even for very high loads, while for Asterisk 1.6 the number of 1280 simultaneous registrations is the highest load it can process satisfactorily. If we use the fair approximation of SIPp clock ticks to milliseconds, we can see on the second chart of the Fig. 3, that for higher loads than 1280 registrations, the registration interval takes up to 14 seconds to complete, which is of course unacceptable. Asterisk 1.8 has no significant issues even for 5 times greater load.

6 Conclusion

In this paper we presented one of many approaches to stress testing of SIP servers, which was made possible by adopting of modular approach to test scenario design. The presented results showed that the new major version of Asterisk PBX was also a major leap in the effectiveness of handling burst loads of Register requests. The decision was made based on the data collected on the client devices exclusively, but thanks to the possibility of collecting data even on the SIP server we can determine that the limiting factor was the Asterisk's ability to successfully and quickly enough process the UDP segments from the UDP buffers.

The sowed example of measurements can also be combined with the call tests or there is a possibility to test not the bursts but the slow sequential increase of the number of simultaneous registrations. In other word the possibilities of the newly redesigned platform are vast.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

References:

[1] D. Malas, A. Morton, Basic Telephony SIP End-to-End Performance Metrics, Internet Engineering Task Force: RFC 6076, January 2011, ISSN 2070-1721.

[2] J. Meggelen, J.Smith, L. Madsen, Asterisk: The Future of Telephony. O'Reilly, (2007)

[3] M. Voznak, J. Rozhon, Performance testing and benchmarking of B2BUA and SIP Proxy, 33rd International Conference on Telecommunication and Signal Processing, Vienna, (2010)

[4] M. Voznak, J. Rozhon, SIP back to back user agent benchmarking, Proceedings 6th International Conference on Wireless and Mobile Communications, ICWMC 2010 , art. no. 5629122, pp. 92-96, Valencia, (2010)

[5] M. Voznak, J. Rozhon, Approach to stress tests in SIP environment based on marginal analysis, Telecommunication Systems, pp. 1-11, ISSN 10184864, DOI: 10.1007/s11235-011-9525-1 (2011)

[6] M. Voznak, J. Rozhon, SIP infrastructure performance testing, 9th WSEAS International Conference on Telecommunications and Informatics, TELE-INFO '10 , pp. 153-158, Catania, (2010)

[7] S. Poretsky, V. Gurbani, C. Davids, Terminology for Benchmarking Session Initiation Protocol (SIP) Networking Devices, Internet Engineering Task Force: RFC Draft v03, (2011)

[8] S. Poretsky, V. Gurbani, C. Davids, Methodology for Benchmarking SIP Networking Devices, Internet Engineering Task Force: RFC Draft v03, (2011)