iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.45035

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Automated Testing of Session Border Controller in Simulated Production Environment

Anirudh Praveen[1], Saurav Kumar[2], Sindhu Rajendran[3]
[1, 2, 3]*Department of Electronics and Communication Engineering, R.V College of Engineering*

Abstract: ***A Session Border Controller plays a crucial role in Unified Communications, Contact Centre Modernization and Cloud communication Services. Therefore, testing the functionality of SBC is extremely critical to the modern communication infrastructure. Testing various call scenarios through the SBC by manually making call through devices is cumbersome, time-consuming and error-prone. Session Border Controller (SBC) devices sit in between a lot of traffic in a service provider or VoIP infrastructure and thus testing the SBC to see if all most relevant call scenarios are tested is a must in today's production SBC environment. To enable an automated tested approach, we have simulated the setup of a real production environment SBC and tested the calls using SIPp scripts to test all the relevant VoIP call scenarios. This paper covers a methodology on how to test Session Border Controllers to be robust and reliable to be deployed in enterprise and service provider customers. This paper demonstrates how to make these simulated calls using automated scripts using an XML (eXtensibke Markup Language) based scripting tool called SIPp. To verify these tests, an opensource and free packet sniffer like Wireshark is used to follow through and check if all the intended calls have taken place. In this paper, we perform thirteen of the most relevant SIP (Session Initiation Protocol) calls by running scripts at the endpoints of our setup. The proposed approach of testing SBC makes the overall process of testing easier through automating via scripts and also allows a novice with basic understanding of the Session Initiation Protocol to test and debug issues related to the SBC faster.***

Keywords: SBC, SIP, SIPp scripts, VoIP, User Agent Client (UAC), User Agent Server (UAS)

I. INTRODUCTION

A Session Border Controller (SBC) are devices that are deployed on the network borders to ensure reliability of communication. SBC is an integral part of the modern Voice over Internet Protocol (VoIP) infrastructure and controls and manages all traffic through a given VoIP infrastructure such as an enterprise VoIP setup [1].

Session Border Controllers are initially deployed within the service provider's network. SBCs ensure that VoIP calls are routed properly between network providers and the end users. SBCs provide security and VoIP call management across different VoIP networks [2]. SBCs provide the functions needed to interconnect carriers and access networks. SBCs have four logical functions namely, signalling function, resource policy function, security function and media function [3]. SBCs also help in lawful intercept of VoIP calls as they are at an ideal location which at the border of two VoIP networks [4].

SBCs use the Session Initiation Protocol (SIP) for creating, modifying and terminating VoIP calls [5]. Once the call is established, the audio/video media transfer is done using the Real-time Protocol (RTP) [6]. Both these protocols are application layer protocols and thus, SBC is essentially a device that operates on the application layer. The SIP protocol is not a secure protocol by itself [7] and needs extra security features to ensure security which the SBC provides. The user agent that initiates the call is known as the user agent client (UAS) and the user agent that receives the call is known as the user agent server (UAS). The SBC deals with a lot of different SIP call scenarios and has to be robust enough to handle all these calls. Thus, extensive testing before deployment is an absolute necessity. But, this sort of testing is time consuming as one has to manually go through all the necessary test cases. Here, we propose a faster and automated way to test the SBC in a simulated production environment using an XML based scripting tool called SIPp [8].

II. METHODOLOGY AND IMPLEMENTATION

The SBC is first setup in the lab with the configurations and setup as close to the production environment as possible. This setup involves load balancers (session router core and session router edge) [9], SBC and two endpoints to simulate the call flows.

The call flow details are sampled from an existing production SBC that has already been previously deployed. Using this network profile, we can use deduce the most relevant calls and test them using automated scripts. These scripts are run in the two endpoints of our setup.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

The session routers (SR) act as load balancers for the system so that any one CPU of the SBC is not under high load during the operation of the SBC. There is an extra session router during the ingress call as shown in Fig. 2. This is to avoid the SBC to overload during and ingress call from endpoint 2. Only the ACK and BYE SIP messages are sent directly to the SBC as these are less frequent messages and this helps in efficiency of the overall call.

SIPp is a free and opensource scripting tool and network traffic generator for the Session Initiation Protocol (SIP). SIPp can also read other custom XML scenario files describing from very simple to complex call flows. SIPp can be used to test various real SIP equipment like SIP proxies, B2BUAs, SIP media servers, SIP gateways, etc. It is also very useful to emulate thousands of user agents calling your SIP system. SIPp can also send media traffic through RTP [4].

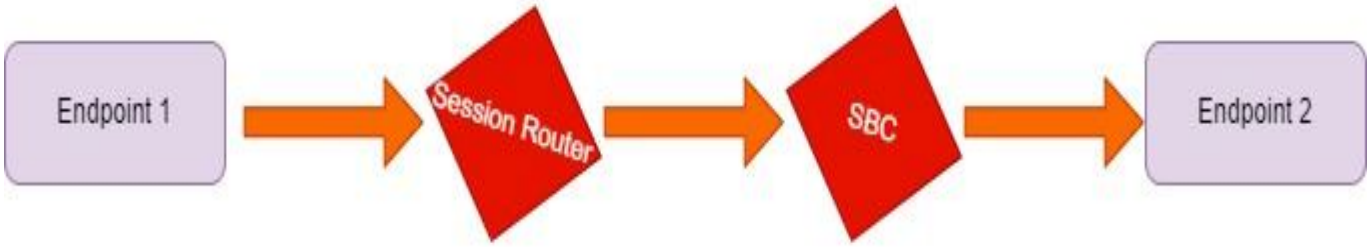


Fig. 3.1 Egress VoIP call from endpoint 1 to endpoint 2 through SBC

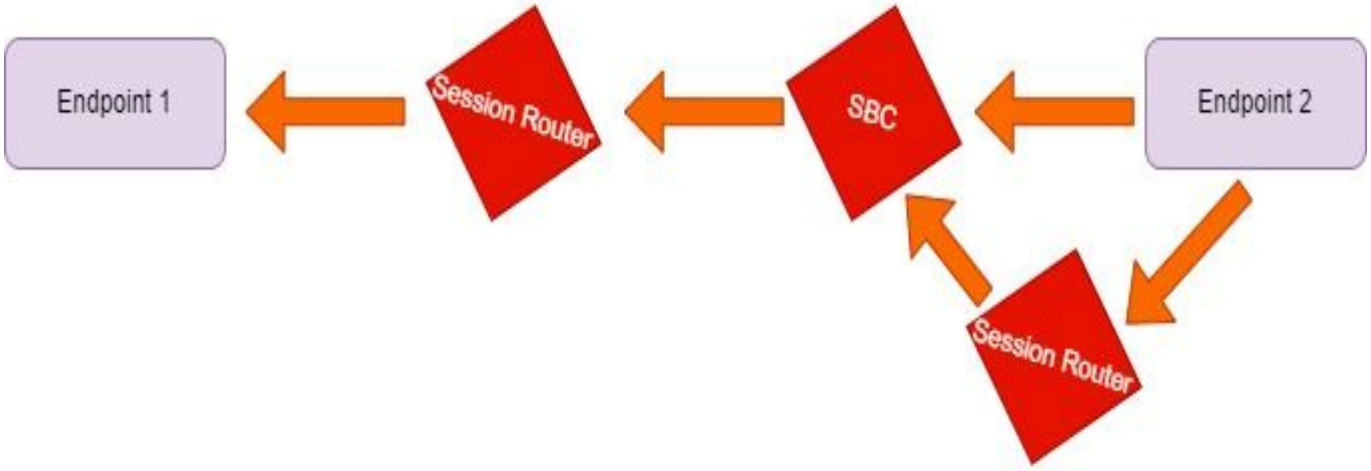


Fig. 3.2 Ingress VoIP call from endpoint 2 to endpoint 1 using SBC

The two endpoints are Linux servers that have SIPp installed. One of these Linux servers acts as a user agent client (UAC) and the other Linux server acts as a user agent server (UAS). The SIPp scripts for both the user agent client and user agent server is written separately for each call scenario. The script for the UAS and UAC is different as each of them expect different SIP headers. One should determine what SIP headers are necessary for a specific call scenario and write the SIPp scripts based on that. This requires a basic knowledge of the Session Initiation Protocol. If there is any audio media involved in the SIPp script, the audio file path must be added in the script. Using the Session Description Protocol (SDP) [10] one can describe how media has to be handled in the SIP calls.

The <recv> tags instruct the user agents to expect messages that are enclosed within the tag and the <send> tag instructs the user agents what to send. The expected SIP responses are also defined in the SIPp scripts.

The Fig 3.3 depicts the process to be followed while writing a SIPp script for a given SIP call scenario.

Both the UAS and UAC scripts must be run simultaneously and tested for different call scenarios based on the most occurring types of calls from a sample of calls that has already occurred in a production SBC system.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

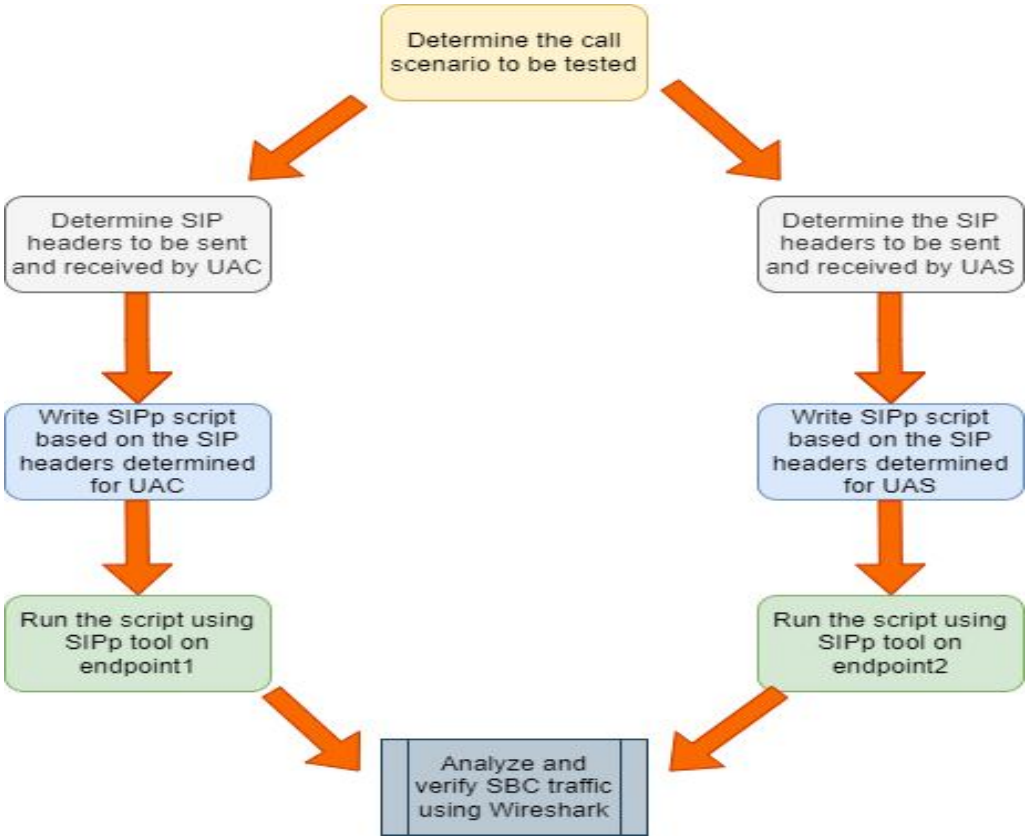


Fig 3.3. Methodology to write SIPp scripts for testing call scenarios

From the data of a similar production SBC system, the different types of calls that are extracted are shown in Table I. This information is extracted from sample of calls that is present on a production SBC.

TABLE I
Call Flow Details

Type	Call Flow Details	% of call flows	No. of messages
Ingress (50% of calls)	1 INVITE with 183 Session Progress and one UPDATE	2.00%	21
	2 INVITE with 183 Session Progress and two UPDATEs	1.00%	25
	3 INVITE with 183 Session Progress and multiple UPDATEs	3.00%	48
	4 INVITE with 183 Session Progress and no UPDATE	70.00%	17
	5 INVITE with 500 internal server Error	1.00%	7
	6 INVITE followed by CANCEL	17.00%	17
	7 INVITE followed by 480	4.00%	4
	8 INVITE followed by 486	2.00%	8
Total % of Ingress calls		**100.00%**	
Egress (50% of calls)	9 INVITE with PRACK	26.00%	21
	10 INVITE without PRACK	25.00%	22
	11 INVITE with CANCEL	19.00%	14
	12 INVITE followed by 486	6.00%	8
	13 reINVITE call flow	24.00%	31
Total % of Egress calls		**100.00%**	

The thirteen calls captured in the network profile is tested using SIPp scripts. All the call almost follows the same format of a basic call flow script, but for every call one can change the SIP headers that the end points send and receive. This allows to one to test all the testcases just by writing scripts and making all the calls required. The figure below shows how a basic script looks like.

III.RESULTS

All the call scenarios are verified by using a packet tracer called Wireshark where all the SIP messages can be seen and a VoIP ladder diagram can be extracted to see if the test passed or failed. This pass or fail is verified by checking if the required SIP messages have reached the intended destination or not. Here, only the SIP signalling messages are checked and not the RTP media packets.

The following VoIP ladder shows how one verifies a test. The Figure 4.1, 4.2 and 4.3 depict the VoIP ladder diagrams from Wireshark. All the rest of test cases follow a similar pattern and one can check if all the messages were going through the SBC and the endpoints.

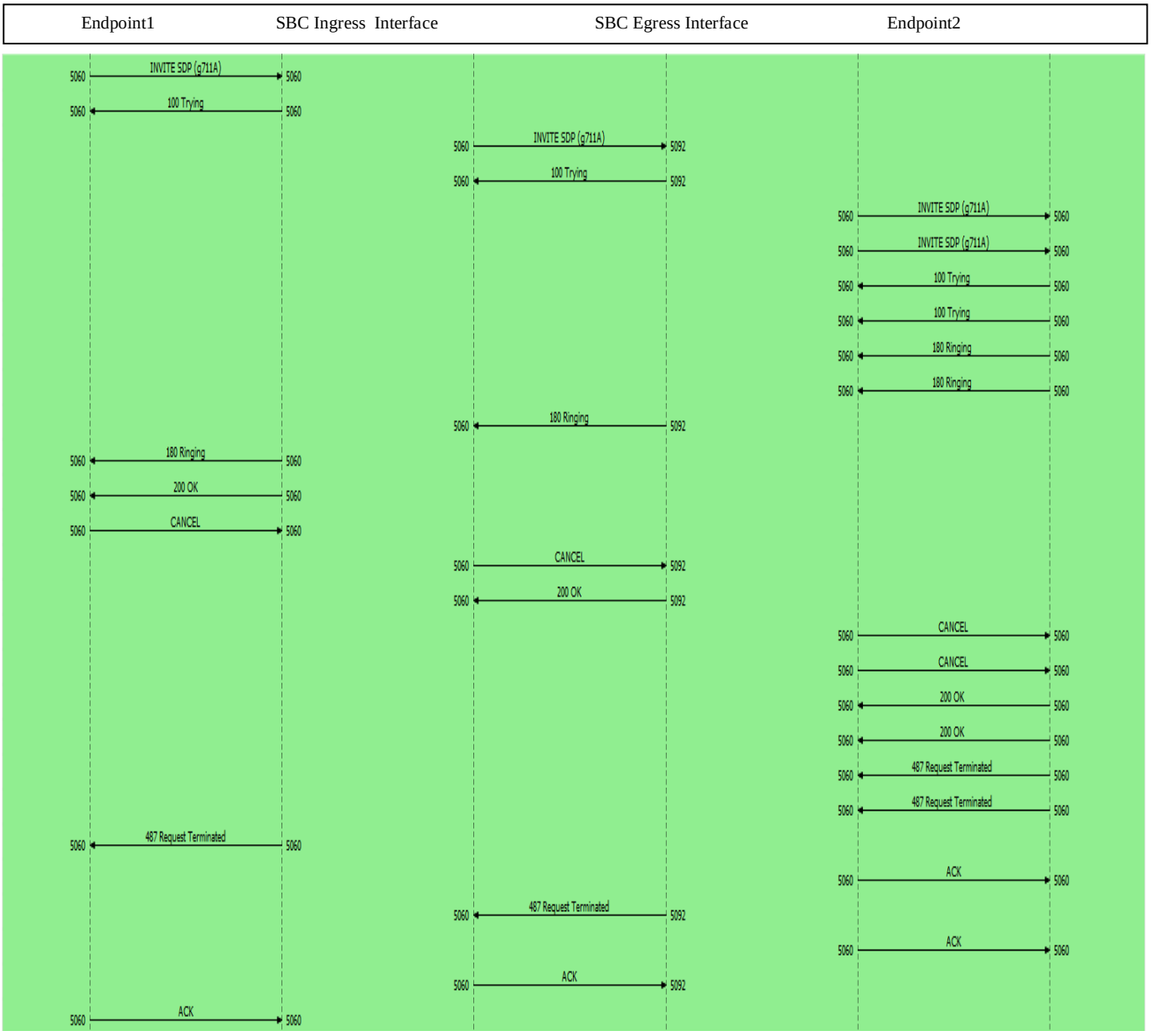


Fig. 4.1 VoIP ladder diagram of test case 11 INVITE with CANCEL

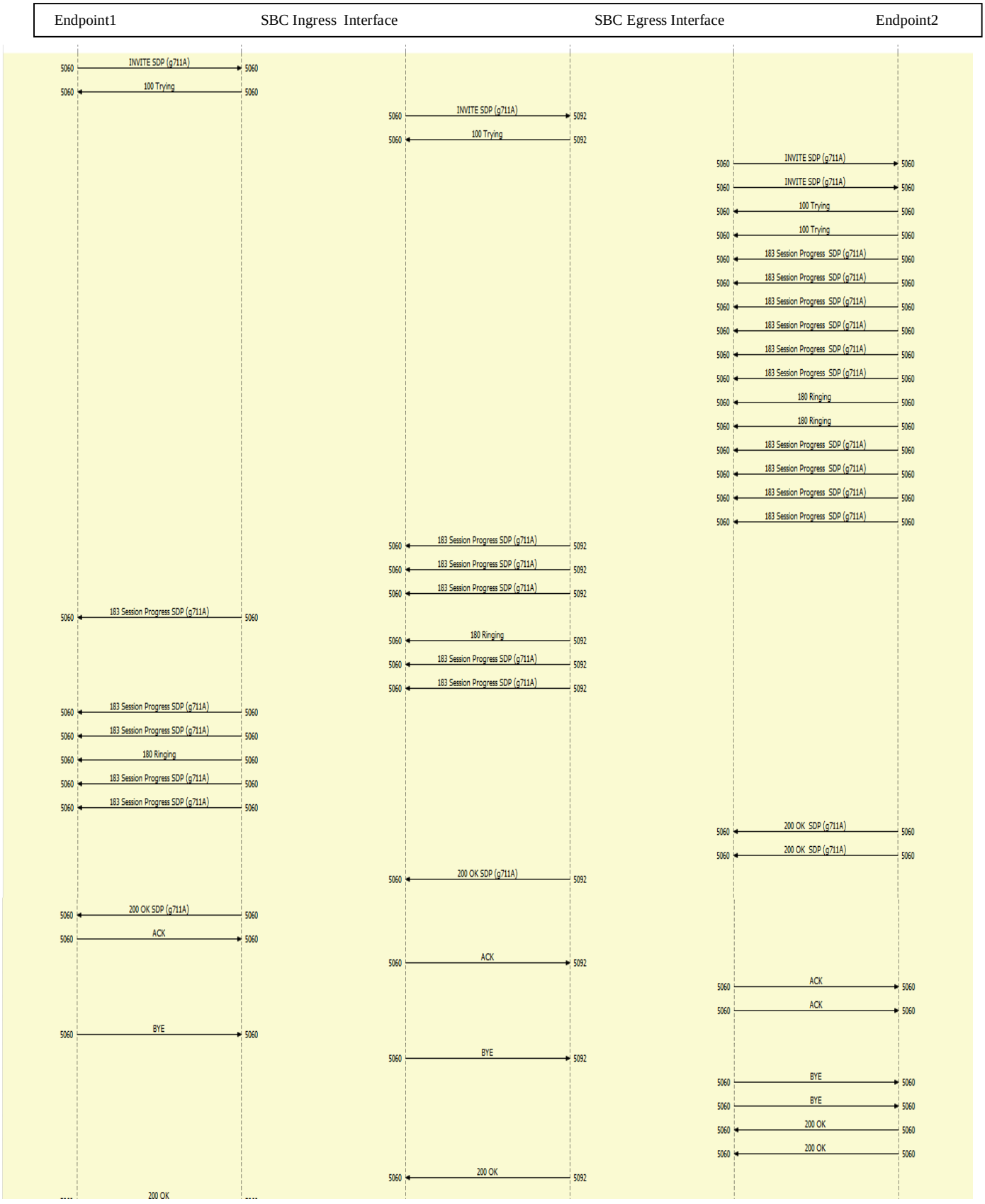


Fig. 4.2 VoIP ladder diagram of test case 10: INVITE without PRACK

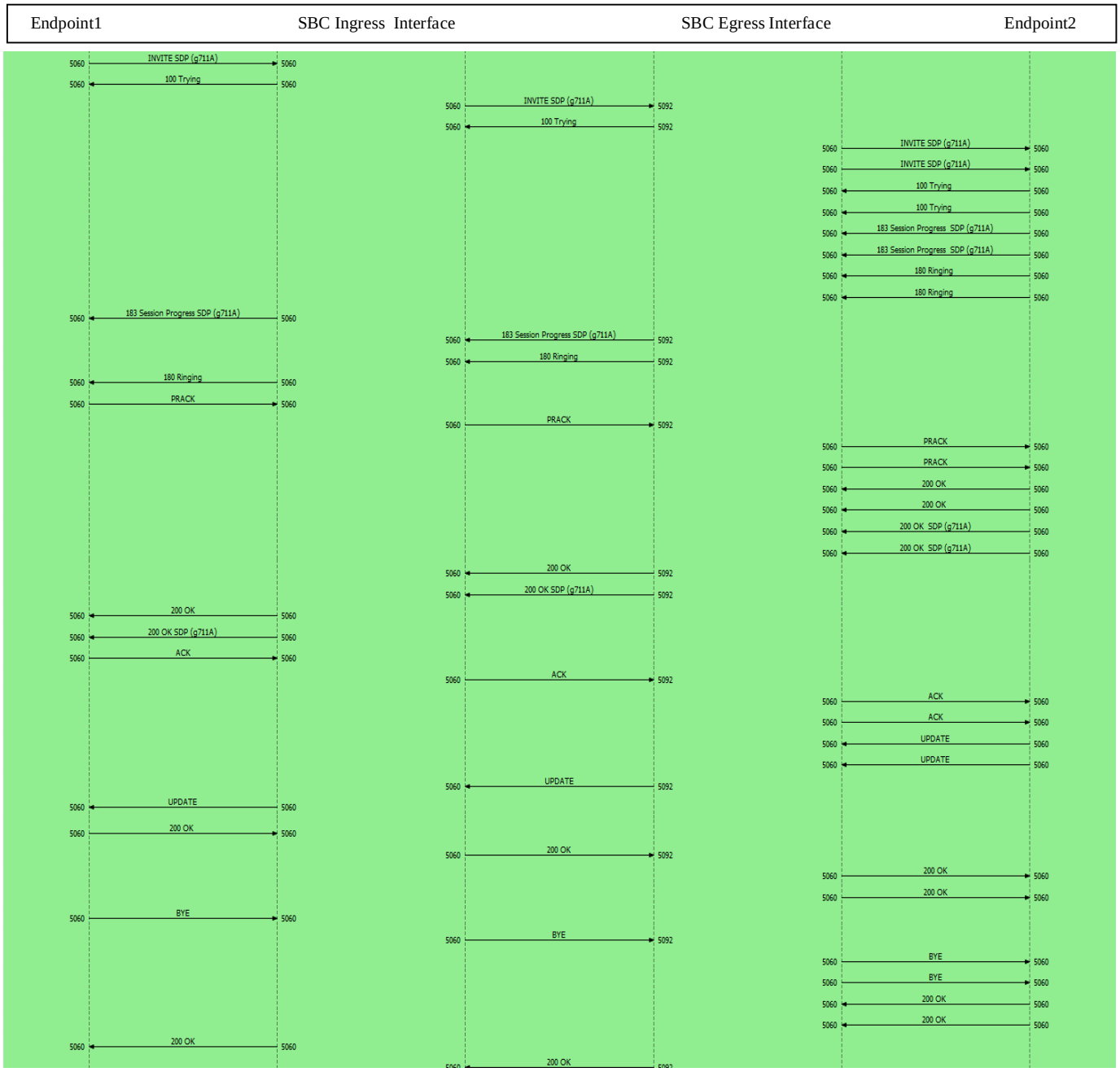


Fig. 4.3 VoIP ladder diagram of test case 9: INVITE with PRACK

Using the above Wireshark VoIP ladder, one can verify all the SIP messages has been sent and the call has behaved as expected by the test. All the other call scenarios follow a similar pattern and is verified using the same way. The SBC ingress and egress interface have two lines respectively and the endpoint is represented by the lines on each of the ends of the VoIP diagram. This form of setup helps the user identify the messages passing through the SBC and help in efficiently verifying tests not just at the endpoints but also within the SBC.

By verifying all the call scenarios from the ladder diagrams, a final result table is created as shown in Table 2. The table represents all the relevant test cases done and the verdict of if the tests have passed or failed. When a test fails, the troubleshooting is done at the SBC and the tests can be run again just by running the scripts at the two Linux endpoints that are setup according to Fig. 3.1 and Fig. 3.2.

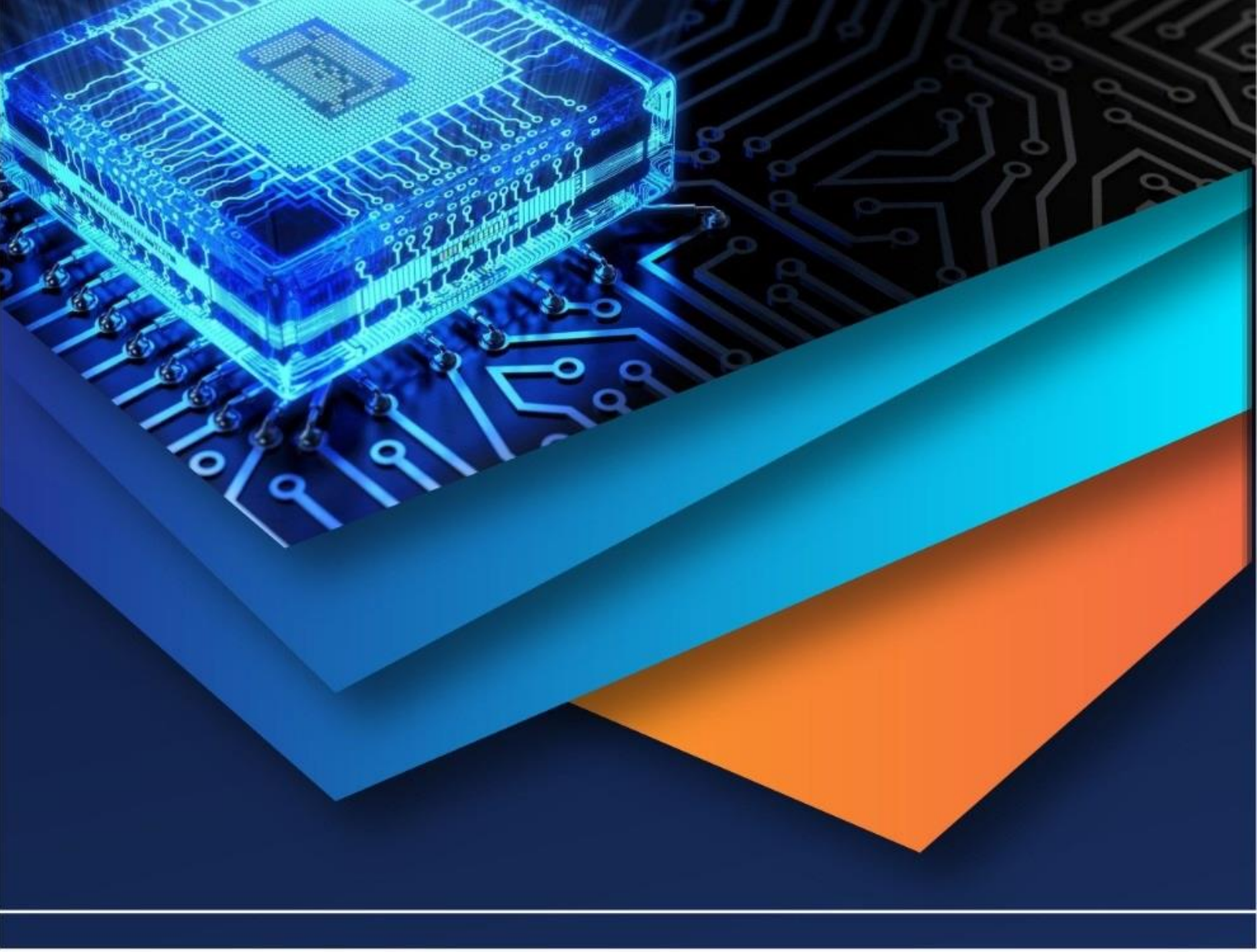
Test Category	Test Case Number	Description	Result (Pass/Fail)
Traffic/Call Flow Verification Testing			
	TC1.1	INVITE with 183 Session Progress and oneUPDATE	Pass
	TC1.2	INVITE scenario with 183 Session in Progress and two UPDATEs.	Pass
	TC1.3	INVITE with 183 Session Progress and Multiple UPDATE	Pass
	TC1.4	INVITE with 183 Session Progress and no UPDATE	Pass
	TC1.5	INVITE with 500 Internal Server Error.	Pass
	TC1.6	INVITE followed by CANCEL.	Pass
	TC1.7	INVITE followed by 480 Temporarily Unavailable.	Pass
	TC1.8	INVITE followed by 486 Busy Here	Pass
	TC1.9	INVITE with PRACK	Pass
	TC1.10	INVITE without PRACK	Pass
	TC1.11	INVITE with CANCEL.	Pass
	TC1.12	INVITE with 486 Busy Here	Pass
	TC1.13	reINVITE call flow	Pass

IV. CONCLUSIONS

The SIPp scripts helped to test all the relevant call scenarios without explicitly making these call scenarios by calling/holding/disconnecting from a device. This in turn saves time for the tester and generates a good result to show to clients to convince them of buying the SBC product as it works best in their production environment scenario. This was done by setting up the SBC according to the client's production requirement and testing the SBC by running the SIPp scripts on the endpoints of the setup. This sort of testing setup can be performed for any SBC and all call scenarios can be tested just by changing the SIPp scripts in the user agent server and the user agent client.

REFERENCES

[1] Lehtinen, M., 2005. Session Border Controller and IP Multimedia Standards.
[2] Cumming, J., 2005. Session border control in IMS.
[3] Hautakorpi, J., Camarillo, G., Penfield, R., Hawrylyshen, A. and Bhatia, M., 2010. Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments (No. rfc5853).
[4] Yang, M., Liu, H. and Li, T., 2010, June. Implementation and performance for lawful intercept of VoIP calls based on SIP session border controller. In 2010 10th IEEE International Conference on Computer and Information Technology (pp. 2635-2642). IEEE.
[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E., 2002. RFC3261: SIP: session initiation protocol.
[6] Lennox, J., Ott, J. and Schierl, T., 2009. Source-specific media attributes in the session description protocol (sdp) (No. rfc5576).
[7] Geneiatakis, D., Dagiuklas, T., Kambourakis, G., Lambrinoudakis, C., Gritzalis, S., Ehlert, K.S. and Sisalem, D., 2006. Survey of security vulnerabilities in session initiation protocol. IEEE Communications Surveys & Tutorials, 8(3), pp.68-81.
[8] Zhang, Y., Clouet, A., Awotayo, O.S., Davids, C. and Gurbani, V.K., 2015, May. Benchmarking the session initiation protocol (SIP). In 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR) (pp. 1-6). IEEE.
[9] Oracle Help Center. (n.d.). Session Router Documentation. [online]
[10] Levin, O. and Camarillo, G., 2006. The Session Description Protocol (SDP) Label Attribute (No. rfc4574).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY