ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

SIP Debugger Tool

[1]Y.V.V.N. Vara Prasad, [2]N.Ravi Kumar, [3]M,Venu Gopal, [4]V.Krishna Pratap, [5]G.Buchi Babu
[1,2]Dept. of CSE, Mother Teresa Institute of Science & Technology, Sathupally, JNTUK, AP, India
[3]Dept. of Science & Humanities, Sri Sarathi Institute of Engineering & Technology, Nuzvid, JNTUK, AP, India
[4,5]Dept. of CSE, Sri Venkateswara Institute of Science & IT, Tadepalligudem, JNTUK, AP, India

Abstract
Day by day, the number of companies interested in IP Telephony are increasing continuously. Session Initiation Protocol (SIP) is one of the most popular VOIP protocol that creates, modifies and terminates associations between Internet end systems, including conferences and point-to-point calls. This paper deals with the design and the implementation of a SIP debugger tool that can be used in the validation process of SIP devices. SIP debugger is a software tool that can be used to verify the compliance of Voice over Internet Protocol (VoIP) devices, such as soft phones to the SIP specifications. Different tools are available on the market to conduct a compliance and interoperability validation phase (Example: Wireshark). However, they often have features limited to packet capturing and decoding. The proposed tool, instead, can be inserted into an SIP network and is capable of observing and analyzing, in an automatic way, the communication steps. It operates by executing three subsequent phases. (1). Capturing and filtering the SIP messages (2). Analyzing them and (3). Comparing the message flow with a set of rules. When verification of some rules fails, an Output is reported by indicating the rule that failed and a list of possible causes.

Keywords
VOIP, SIP, IETF RFC 3261

I. Introduction

Over the past few years, the evolution of the Internet and other related technologies has made it possible to face new challenges. Different protocols have even been standardized for conference services based on the Internet. SIP started as a real time communication protocol for voice over IP but has later on been expanded with new features such as video. SIP is an application-layer protocol that can establish, modify and terminate multimedia sessions or calls. SIP allows the establishment of sessions between two user agents (UAs). A UA contains both a client application that sends SIP requests and a server application that accepts them. SIP is also responsible for basic call management. This includes the ability to initiate a call, terminate a call, and to add and remove users from a call. This protocol can be used to initiate multicast or unicast media sessions. It works together with the Session Description Protocol (SDP), which is in charge of describing the session to be opened. This paper explains the basic ideas and the implementation of the SIP Debugger Tool, as well as the presentation of the test cases. The aim is to create a smart automatic system for the validation and testing process of the interoperability between SIP equipment and compliance with IETF RFC 3261. This goal could also give the possibility to carry out SIP testing to unskilled operators.

Different protocol analyzer tools were already available in the market that will perform real time packet capturing & filtering (Examples: WireShark, Etherel, TCP Dump etc.). The tool proposed in this paper works similary to protocol analyzers because it allows capturing and decoding a live SIP communication between two endpoints. However, it overcomes the limits of those tools. In fact, it is capable of automatically identifying the faults present in the communication and suggesting a list of possible causes.

Due to the capability of automatically identifying the faults of the communication, the SIP automatic debugger tool allows a reduction in the cost of the validation phase because it does not require the intervention of a protocol expert. Furthermore, it provides performances independent of operator skill.

II. SIP Overview

SIP is an ASCII-based (text-based) application layer control protocol. Like other VoIP protocols, SIP can be used to establish, maintain, and terminate calls between two or more endpoints. In fact, SIP supports unicast, mesh, and multicast conferences, as well as combinations of these modes, and implements services such as call forwarding and transfer, placing calls on hold, camp-on, and call queuing by a small set of call handling primitives. SIP is a client–server protocol and is based on different messages exchanged between UAs. Typically, an SIP endpoint is capable of functioning as both a UA Client (UAC) and a UA Server (UAS), but it functions only as UAC or UAS per transaction. From an architecture standpoint, the physical components of an SIP network can be grouped into two categories, i.e., clients and servers.

SIP clients include the following.

- Phones that can act as either UAS or UAC.
- Access gateways provide call control.

SIP servers include the following.

- Proxy servers that are intermediate devices receiving SIP requests from a client and then forwarding the requests on the client's behalf.
- Redirect servers provide the client with information about the next hop or hops that a message should take.
- Registrar servers process requests from UACs for the registration of their current location. Registrar servers are often colocated with redirect or proxy servers.

Basic SIP call management is achieved with a small number of messages called methods. Table I supplies a description ofthe six types of request methods defined by RFC 3261.The messages can be carried by either User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) packets. When a UAC desires to initiate a session, it formulates an INVITE message request. The INVITE request asks a server to establish a session. The INVITE request may be sent either directly to the called party or to a proxy server who is in charge of forwarding the message to the called party. These UASs will frequently need to query the user about whether to accept the invitation. After some time, those UASs can accept the invitation by sending a 2xx response. If the invitation is not accepted, a 3xx, 4xx, 5xx, or 6xx response is sent, depending on the reason of the rejection (Table II). Before sending a final response, the UAS can also send provisional responses (e.g., 1xx message) to advise the UAC of progress in contacting the called user. The response message is composed of a heading line and some body lines. The first line (heading line) is the Status-Line, consisting of the protocol version, followed by a numeric Status-Code, which is a three-digit integerresult code

that indicates the outcome of the attempt to understand and satisfy the request. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automatic tools, whereas the Reason-Phrase is intended for the human user. The first digit of the Status-Code defines the class of response. The last two digits do not have a categorization role. Users in a SIP network are identified by unique SIP addresses. A SIP address is similar to an e-mail address and can be expressed in different forms, e.g., "user@domain," "user@host," "user@ IP_address," or "telephone- number@gateway." Users register by sending a REGISTER request with their assigned SIP addresses to a registrar. A registrar acts as the front end to the location service for a domain, reading and writing mappings based on the contents of REGISTER requests. This location service is then typically consulted by a proxy server that is responsible for routing requests for that domain. Over time, an SIP end user might move between end systems. The location of the end user can be dynamically registered with the SIP server.

Table 1: SIP Request Methods

Sip message	Description
INVITE	Invites a user to a call
BYE	Terminates a connection between users or declines a call
OPTION	Solicits information about a server's capabilities
ACK	Used to facilitate reliable message exchange for INVITEs
CANCEL	Terminates a request, or search, for a user
REGISTER	Registers a user's current location

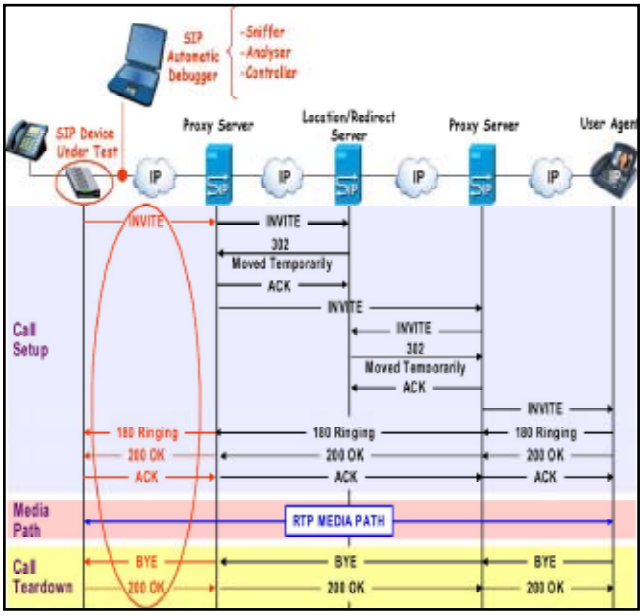


Fig. 1: SIP to SIP Call Flow. The SIP Automatic Tcaptures Outgoing and Incoming SIP Messages of the Home Access Gateway Under Test

Table 2: SIP Response Methods

Response Class	Description
1xx	Provisional responses
2xx	Responses are positive final responses
3xx	Responses are used to redirect a caller
4xx	Client errors
5xx	Server errors
6xx	Global failures

III. SIP Automatic Debugger

The SIP debugger tool was created in Java Programming language using J2SE development environment to run on Windows/Linux operating system. The tool operates on an SIP UA which is a soft phone, by using approach as shown in fig. 2.

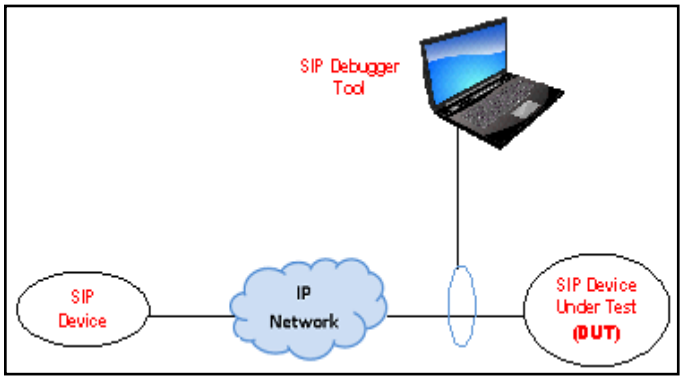
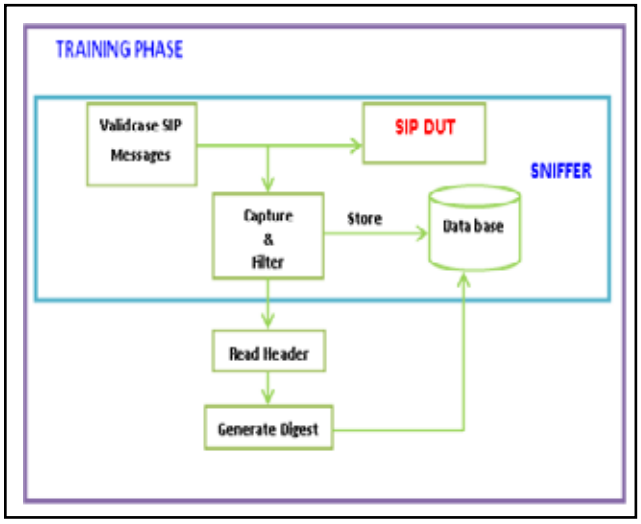


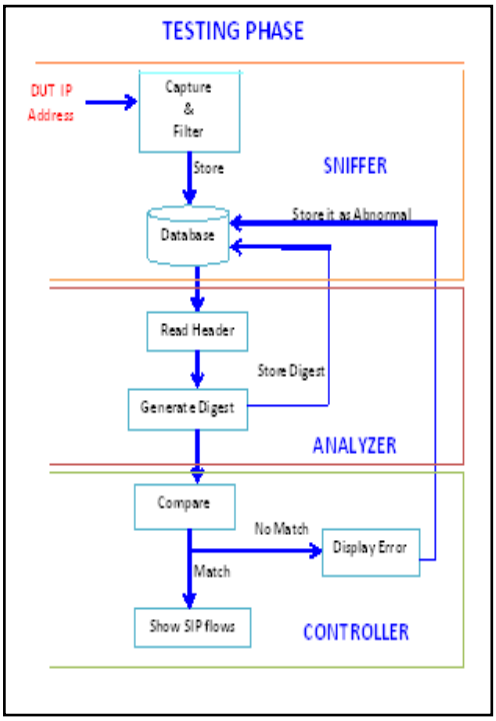
Fig. 2: Scheme of Operation of SIP Debugger Tool

The tool verifies all incoming and outgoing SIP messages directed to or coming from such a DUT and automatically generates detailed reports for anomaly packets.



(a).

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)



(b).

Fig. 3: Proposed Architecture of SIP Debugger Tool

The proposed tool operates in 2 phases

1. Training phase
2. Testing phase

1. Training Phase

During training phase, the tool is trained with a valid set of SIP messages. These SIP messages are those that will not crash the DUT and were according to SIP RFC 3261 specifications. They were captured by the tool and stored in a database. The Read header function obtains the header of the SIP message. The Generate digest function obtains digest by comparing the header with 104 predefined set of SIP headers. If there is a match the corresponding bit is set to 1 in digest. Otherwise, the bit is set to 0. Thus a 104 bit message digest is obtained for each valid SIP message and is stored in the database.

2. Testing Phase

During testing phase, the system is tested by sending some non trusted SIP messages. All the SIP messages directed to the DUT were captured and filtered by the Sniffer component of the tool & were stored in the database. The Analyzer component is responsible for removing the header of the SIP message and then generates digest using the mechanism explained earlier. The Controller component compares the digest generated for each message with the digest of the valid case messages that were already stored during the training phase. If no match is found an error message is displayed stating that, it is an abnormal packet. All such abnormal packets were stored in the database as abnormal messages. These abnormal packets are those that were not previously observed during testing phase and may cause the DUT to crash. A report is generated consisting of only abnormal packets. The testers can then reproduce or resent those abnormal packets to observe the reason for crash of the DUT

III. Experimental Test Methodology

The methodology adopted in conducting the validation tests on the SIP debugger tool consists of selecting some SIP test cases, executing them and analyzing them. The generated report consisting of abnormal packets is shown in fig. 4

When these abnormal packets were resent, it is observed that the DUT is crashed for some of these anomaly packets that were stored by the tool.



Fig. 4: Report Showing the List of Abnormal Packets

IV. Conclusion

In this paper, an SIP automatic debugger tool designed to be used in the test and validation process of SIP devices has been presented. The tool allows the analysis of SIP conversations (offering additional features for the automatic detection of faults) that occurred during the communication. These features provide significant advantages in terms of optimizing the effort involved in SIP product testing. Independent from the operator's skill, the SIP automatic debugger tool makes it easier to analyze the test results, to identify and fix eventual bugs, and to read the generated reports.

References

[1] D. Bao, D. L. Carnì, L. De Vito, L. Tomaciello,"Session initiation protocol automatic debugger", in Proc. IEEE Instrum. Meas. Technol. Conf., Warsaw, Poland, May 2007, pp. 1–6.
[2] [Online] Available: http://www.wireshark.org
[3] [Online] Available: http://www.linkbit.com
[4] [Online] Available: http://ant.comm.ccu.edu.tw/sip
[5] [Online] Available: http://www.sipcenter.com/sip.nsf/html/Sponsors+GL+Communications
[6] [Online] Available: http://www.empirix.com/products-services/v-hca.asp
[7] [Online] http://Available: http://sipp.sourceforge.net
[8] [Online] Available: http://www.sipcenter.com/sip.nsf/html/Sponsors+Radvision
[9] [Online] Available: http://www.iptel.org/ser
[10] [Online] Available: http://www.metalinkltd.com/downloads.php
[11] [Online] Available: http://www.sipcenter.com/sip.nsf/html/Compliance+Engine
[12] [Online] Available: http://www.sipsak.org



Y V V N Vara Prasad received his M.Tech form JNTU Hyderabad, AP, India in the year of 2009. M.Tech (IT) (Punjab U niversity)in the year of 2003. M.Sc from Gulbarga University in the year of 1990. He is a member of IETE and life member of ISTE. He had 21 years of teaching experience and good academician. Present he is working as an Assistant Profressor in IT Department in Mother Theresa institute of Science and Technology.



N.Ravi Kumar Re received his M.Sc in mathematics from Acharya Nagarjuna University, Guntur, AP, India in the year of 2005. He did his M.C.A from JNTU Hyderabad in the year of 2009. Present he is studying M. Tech from JNTU Hyderabad. His research interests in graph theory, data structures, JAVA and Auto mata theory.



M.Venu Gopal received his M.Sc., in mathematics from Acharya Nagarjuna University, Guntur, AP, India in the year of 2005. He did M.Phil in the Specialization of Algebra in mathematics from Andhra University, Vishakhapatnam, AP and India. He has been working as Asst. Professor of mathematics & HOD S & H in the Dept of Basic Science and Engineering in Sri Sarathi Institute of Engineering and Technology; Nuzvid, AP and India since 2006. His research interests in Graph Theory, Data Structures, Lattice Theory, Formal Language, Auto mata Theory.



Mr. V. Krishna Pratap received his M.Tech , in CSE from JNTU KAKINADA University , AP and India in the year of 2011. He studied B.Tech in the Specialization of Information Technology From JNTU Kakinada University, AP and India. He has been working as Asst. Professor in the Computer Science and Engineering in Sri Venkateswara Institute of Science And Information Technology; Tadepalligudem, AP, India. His Research Interests In Computer Networks, Data Structures, Information Security, Mobile Computing.



Mr.G.Buchi babu received his B..Tech, from JNTU KAKINADA University, AP and India in the year of 2009 . He studied B.Tech in the Specialization of Information Technology from JNTU Hyderabad University, AP, India. He has been working as Assistant. Professor Dept in the Computer Science and Engineering in Sri Sarathi Institute of Engineering and Technology; Nuzvid, AP, India since 2009. His research interests in Data Mining, Data Structures, Formal Language and Automata Theory.