

The SIPv6 Analyzer Demo Scenario v1.0

Whai-En Chen, Yueh-Hsin Sung, and Chang-Yu Tsai

Department of Computer Science and Information Engineering

National Chiao Tung University

{wechen,yhsung,cytsai}@csie.nctu.edu.tw

I. Introduction

Some famous packet analysis tools such as Ethereal [5] and Sniffer [6] can parse various protocols, set the filtering rules and provide the statistic of the networks. For *Internet Protocol version 6* (IPv6) [4] *Session Initiation Protocol* (SIP) [1] -based *Voice over IP* (VoIP) environments, providing basic protocol analysis is not enough. On the other hand, the commercial VoIP analysis tools costs too expensive and does not provide all functions that we need. In *National Chiao Tung University* (NCTU) VoIP Laboratory, we base on open source WinPcap library [7] and Ethereal parser to develop an IPv6 SIP Analyzer called *SIPv6 Analyzer* to assist the development and deployment of the IPv6-based SIP applications.

The SIPv6 Analyzer not only parses 512 protocols (the same as Ethereal) but also provides IPv6 SIP and *Real-time Transport Protocol* (RTP) [2] functions for SIP-based applications without extra configuration. To provide the convenient functions, the SIPv6 Analyzer provides the SIP viewer, the RTP Spy and the statistics. The SIP viewer groups the SIP messages into several sessions according to the SIP dialogs/call-legs (i.e., SIP Call-ID, From, and To header fields), and generates the SIP message flowcharts according to the source and destination IPv6 addresses in basic mode and SIP header fields (e.g., the SIP *Via* header field) in advanced mode. These flowchart functions are convenient tools for the users to quickly understand the SIP message transmission. The RTP viewer collects RTP packets according to the media description in the *Session Description Protocol* (SDP) [3] fields and

groups them into the RTP sessions according to the SSRC field of the RTP header. The SIPv6 Analyzer provides the RTP playback function that can play both voice and video media contained in the captured RTP packets. To sum up, the SIPv6 Analyzer is useful for the development and deployment of the SIP-based applications. The following sections present the detail steps of a demo scenario that utilizes the SIPv6 Analyzer to analyze an IPv6 SIP based VoIP communication. The user can follow the steps to obtain the same results that we exercised in NCTU VoIP Laboratory.

II. Experimental Environment :

Figure 1 illustrates an experimental environment of the SIPv6 Analyzer and the parameters of each component. The experimental environment includes an IPv6-enabled SIP Server, two *IPv6 SIP User Agents* (SIPv6 UAs, i.e., UA1 and UA2) and the SIPv6 Analyzer. The SIP Server contains a SIP registrar and a SIP proxy, and its IPv6 address is **2001:238:f88:a::15**. Both SIP UAs register their IPv6 addresses to this IPv6 SIP server and communicate with each other through this SIP server. Both UA1 and UA2 are software-version VoIP Phones and developed by NCTU VoIP Laboratory. The IP address of UA1 is **2001:238:f88:310:204:acff:fe8b:1ff**, and the IP address of UA2 is **2001:238:f88:310:60d:20ff:fe13:86f2**. To capture the packets that pass through UA2, the SIPv6 Analyzer should be installed on the same subnet as UA2. In this example, the SIPv6 Analyzer, which is installed on the same hub as UA2, captures SIP messages and RTP streams between UA1 (the calling party) and UA2 (the called party).

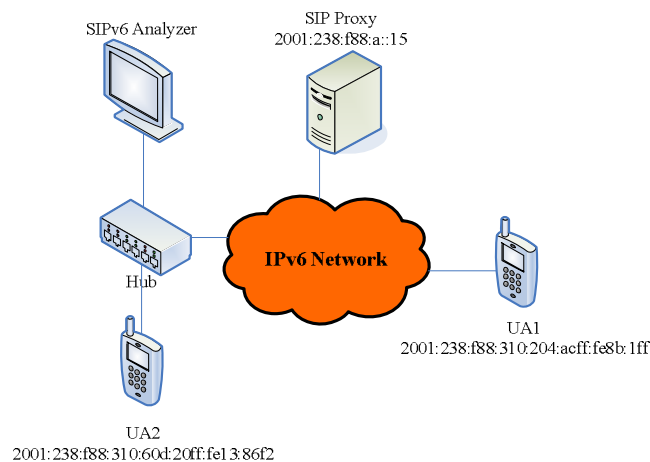


Figure 1. Experimental environment

All the captured packets in this example are stored in a file (i.e. IPv6sample.cap) and can be obtained at <http://www.csie.nctu.edu.tw/~yhsung/IPv6sample.cap>. By using the **Open Offline Packet** function of SIPv6 Analyzer, the user can restore the captured packets and see the same results as that presented in this proposal. The detail description of the installation and the SIPv6 Analyzer functions can be found in the user guide document (i.e. user-guide.doc).

III. IPv6 Packet Analysis :

To capture the packets between UA1 and UA2, the user can click **Local Capture** button (see Figure 2(1)). Then the user will see the **Local Capture** dialog. The user may change **Analysis Name** (see Figure 2(2); the default value is **Project n** where n increases automatically) for each project. To open a project, the user should select **Available Interface(s)** (at least one interface; see Figure 2(3)), then add the interface(s) into the **Selected Interface** list (see Figure 2(4)) and finally click **OK** button (see Figure 2(5)).

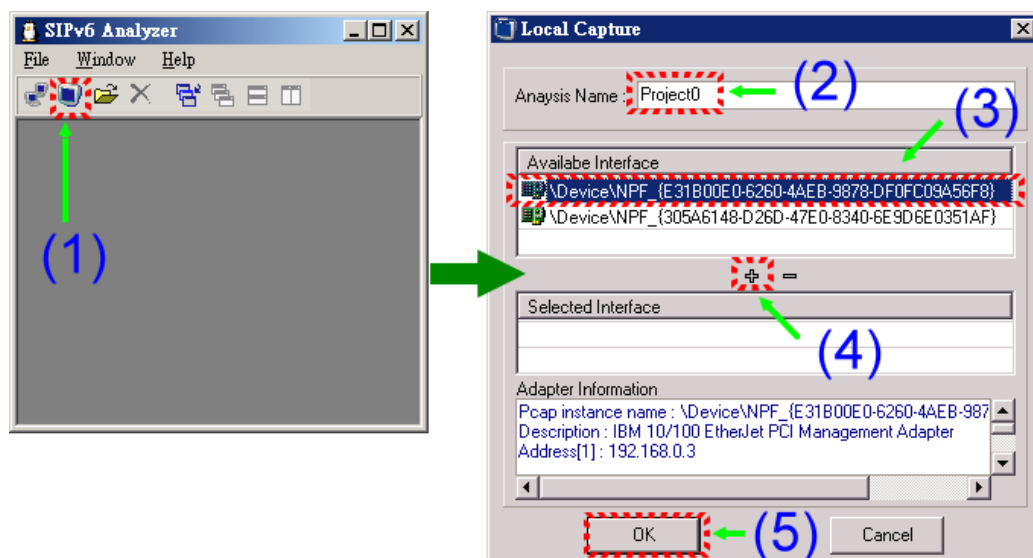


Figure 2. Open a project

After opening the project, the user can start to capture the packets. First, the user can click **Start/Stop capture** button (Figure 3(1)) to start or stop packet capturing. Then the user can select the **Packet Viewer** tab (Figure 3(2)) and see the number of **Captured** packets in the

status bar (Figure 3(3)) is increasing. To see the detail protocol fields, the user can select an entry in **Frame List** (Figure 3(4)), double-click to expand the tree (Figure 3(5)), and then see the **Detail Frame information** (Figure 3(6)) of the packet. In Figure 3(7), the **Raw Data** of the selected packet is dumped in **Hex mode**.

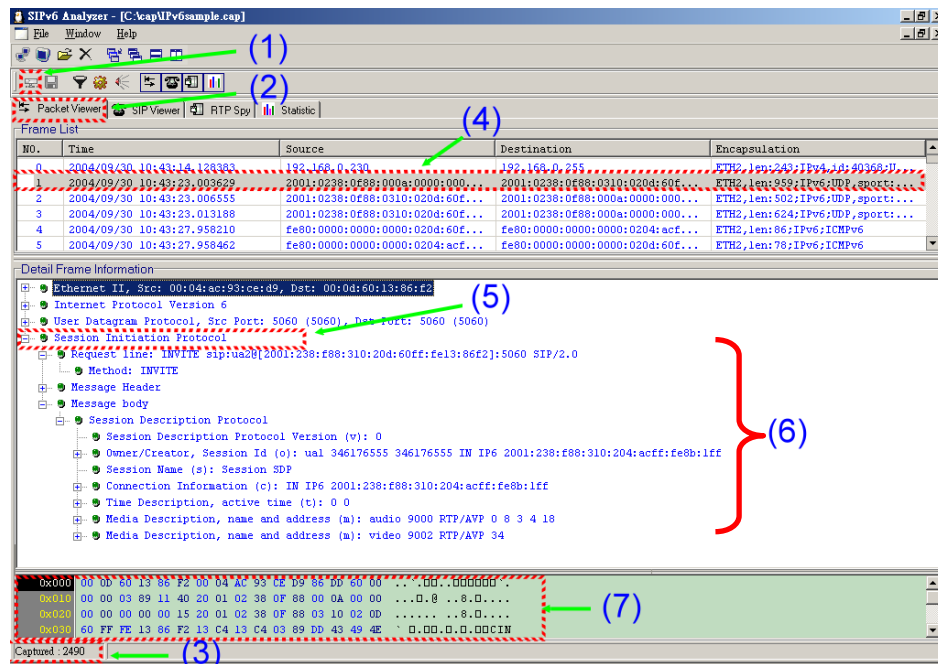


Figure 3. An Example of Packet Viewer

IV. SIP Message List and Flowchart :

To investigate the SIP signaling protocol, the user can select the **SIP Viewer** tab (Figure 4(1)) and find a SIP session shown in the **Dialog List** (Figure 4(2)). When the user selects an entry in **Dialog (Call-leg) List**, then all SIP messages belonging to the SIP session are shown in the **SIP Packet List** (Figure 4(3)). Each item in the SIP Packet list contains the sequence number, the captured time, the SIP Request-URI and IP addresses. If the user attempts to see all SIP header fields, he can select a SIP message and **Double-click** to expand the tree (Figure 4(4)). In a real SIP VoIP network, there may be more than one SIP UAs initiate the SIP INVITE transactions at the same time. Therefore, the SIPv6 Analyzer may simultaneously receive the interleaving SIP messages. In this situation, the user hardly analyzes the SIP protocol since the SIP messages are not displayed in order. The SIP viewer, which merges SIP messages

according to the SIP dialogs, provides a convenient interface to investigate the SIP messages.

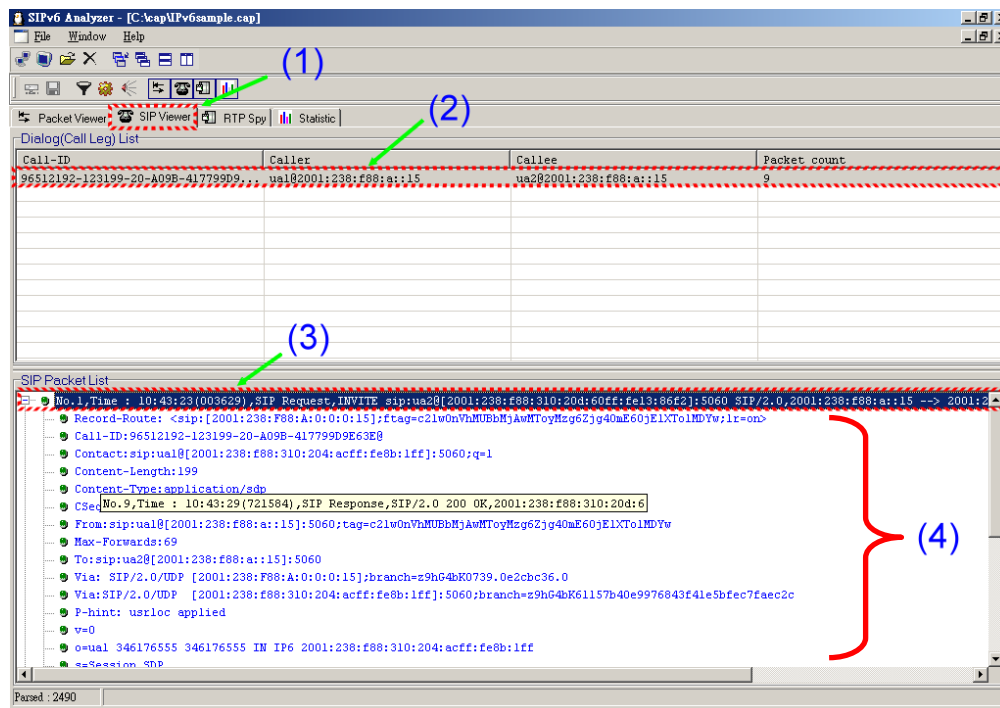


Figure 4. SIP Message List

To generate the SIP flowchart, the user can select one entry(a SIP session) in **Dialog (Call-leg) List** (Figure 5(1)) and click the **Right button** of the mouse. Then the user can see the popup menu. The user can select **Draw Flowchart** item to open the SIP flowchart in basic mode (Figure 5(2)) or select **Draw Flowchart from header** to open the SIP Flowchart in advanced mode (Figure 5(3)).

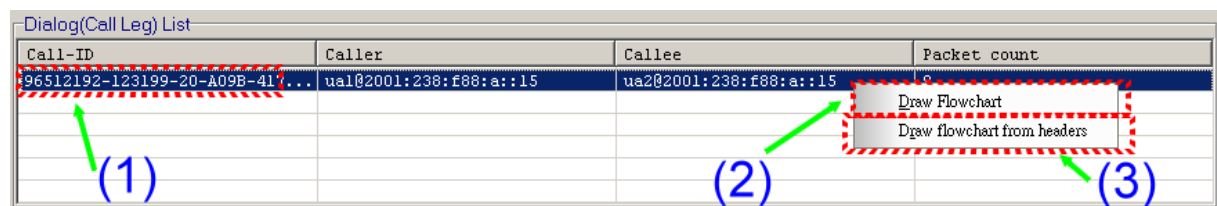


Figure 5. SIP flowchart startup sequences

Figure 6 shows the basic mode of the SIP flowchart where the **Blue Line** represents the SIP **Request** message (Figure 6(1)) and the **Green Line** represents the SIP **Response** message (Figure 6(2)). This flowchart is generated by using the source and destination IPv6 addresses. In this example, the ACK message is directly sent from the UA1 to the UA2 after the UA1

obtains the current IP address of the UA2 from the 200 OK message. This message seems strange in the basic mode because the user does not know the existence of UA1. In this case, the user is suggested to open the flowchart in advanced mode.

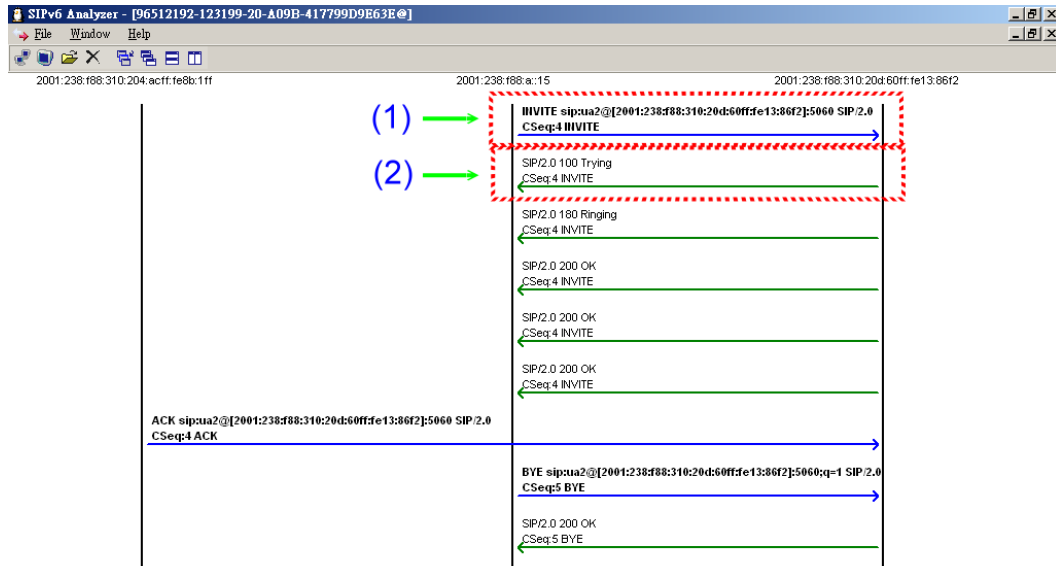


Figure 6. SIP Flowchart (in basic mode)

Selecting **Draw Flowchart from header** item in Figure 5(3), the SIPv6 Analyzer generates the SIP flowchart in advanced mode according to the *Via* header fields of the SIP messages. In advanced mode, the SIPv6 Analyzer draws the predictive SIP messages by using dashed lines. In this flowchart, the **Blue dashed-line** (Figure 7(1)) presents the SIP request message and the **Green dashed-line** (Figure 7(2)) presents the SIP response message. In this mode, the user can see all devices (i.e., UA1, UA2 and SIPv6 server) involved in this communication. Therefore, the ACK message does not seem strange any more.

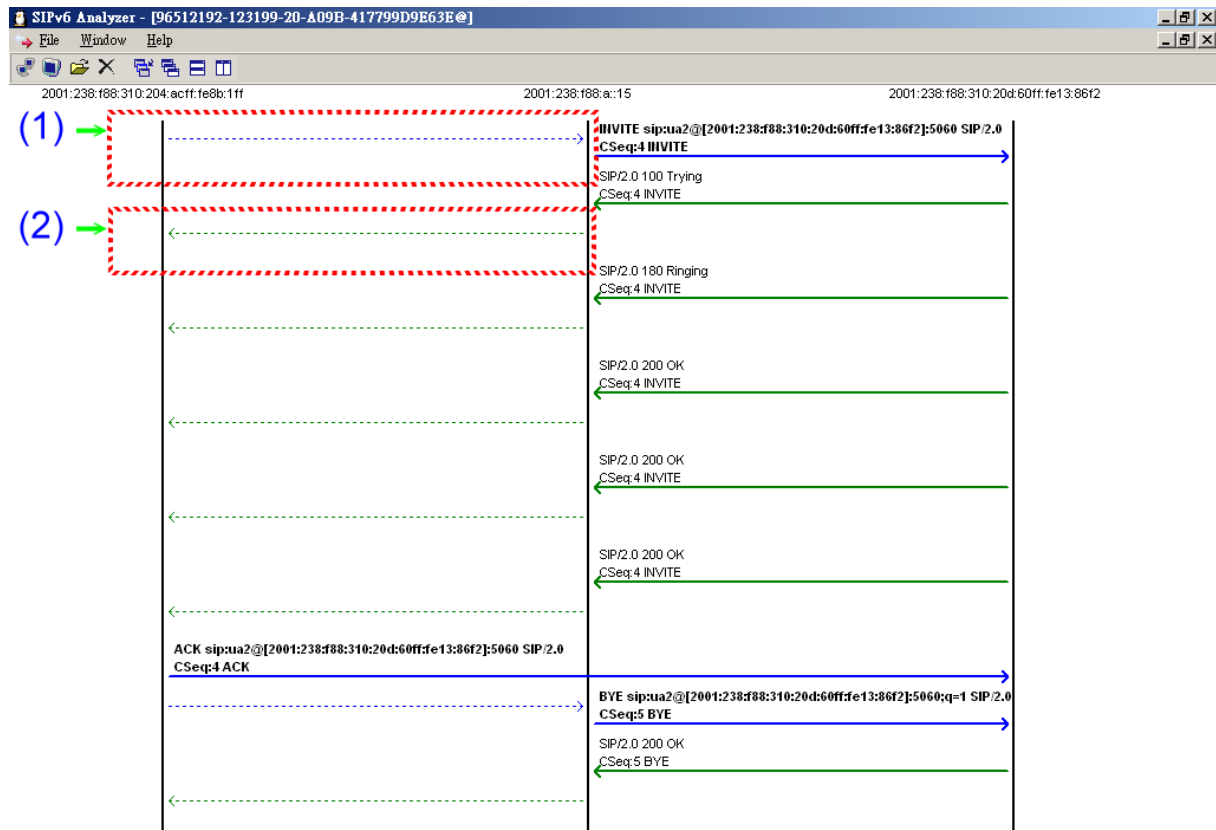


Figure 7. Advanced SIP Flowchart

V. RTP playback :

The RTP playback function of the SIPv6 Analyzer collects the RTP packets into several RTP streams according to the SDP fields. The user can select **RTP Spy** tab (Figure8(1)). The RTP streams, which contain the **Session** (destination IP/port), **SSRC** field, **Media Type**, **Packet Count**, and **Length** information, are listed in the **Session List**. To play back the video/voice, the user can select an entry (Figure8(2)) in **Session List** and **double-click** to add this entry to **Media Instance**. Then the user can select the entry in **Media Instance** and click the **Play/Hold/Stop** button (Figure8(3)) in the control panel to control the playback of the selected RTP stream.

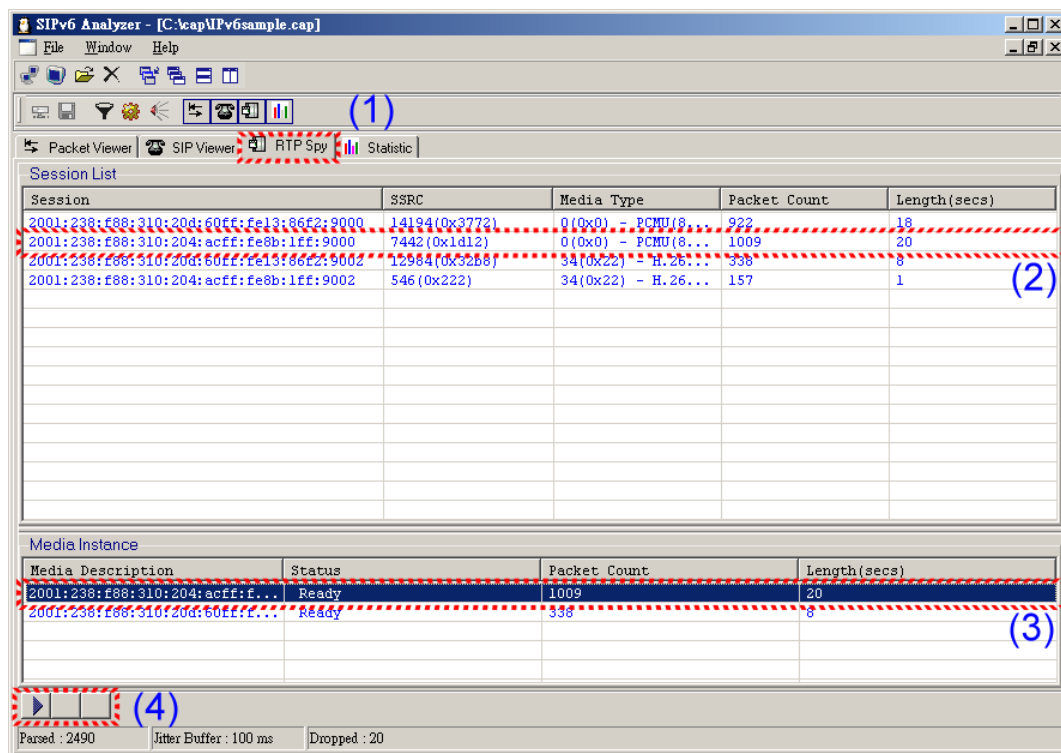


Figure 8. RTP playback

VI. Conclusions

This document provides a demo scenario that analyzes the IPv6 SIP-based VoIP network through the SIPv6 Analyzer. We hope that the users can learn how to operate the SIPv6 Analyzer through exercise this demo scenario. More information of a life demo video can be found in the web site (http://www.csie.nctu.edu.tw/~yhsung/sipv6_analyzer).

Reference

- [1] RFC 3261. SIP: Session Initiation Protocol. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. June 2002
- [2] RFC 3550. RTP: A Transport Protocol for Real-Time Applications. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. July 2003
- [3] RFC 2327. SDP: Session Description Protocol. M. Handley, V. Jacobson. April 1998
- [4] RFC 2460. IPv6: Internet Protocol, Version 6 Specification. S. Deering, R. Hinden. December 1998
- [5] Ethereal. <http://www.ethereal.com>
- [6] Sniffer. <http://www.sniffer.com>