Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp756-761)

NTP VoIP Platform: A SIP VoIP Platform and Its Services[1]

Whai-En Chen, Chai-Hien Gan and Yi-Bing Lin
Department of Computer Science
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu,
Taiwan, R.O.C., 300

Abstract: - This paper introduces the major components of a SIP-based VoIP platform, referred to as NTP VoIP platform. Based on the NTP VoIP platform, the researchers can develop and deploy their applications and services. For lawful interception, we propose a monitoring system that provides call detail records and interception function. We also propose a conference system for audio and video conferences. In this paper, we provide detailed message flows to show how the monitoring system and the conference system work.

Key-Words: - **CDR (Call Detail Record), LI (Lawful Interception), VoIP, SIP, RTP**

1 Introduction

Many applications have been developed for *Internet Protocol* (IP) networks. Among them, Voice over IP (VoIP) is one of the most important applications. *Session Initiation Protocol* (SIP), which supports functions to integrate instant messaging, user presence, and multimedia communications [1], is the most popular signaling protocol for VoIP call control. Under the *National Telecommunication Development Program* (NTP), we have established a VoIP platform referred to as NTP VoIP platform that allows researchers and students to develop and deploy SIP-based VoIP applications.

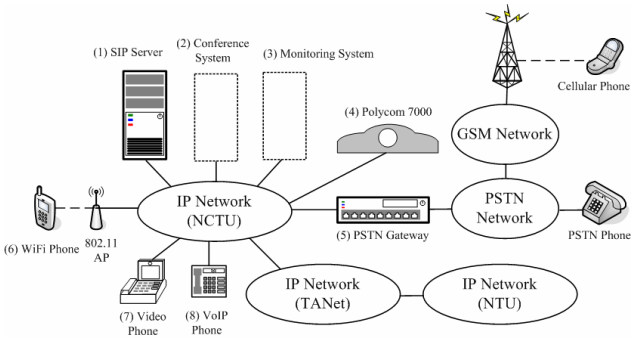

Figure 1. NTP VoIP Platform

Figure 1 illustrates the NTP VoIP platform architecture. The components are described as follows.

- **SIP server** (Figure 1(1)) provides primary capabilities for call-session control in NTP VoIP platform. A SIP server processes SIP request and response messages as a SIP proxy. It also plays a role as a SIP registrar to store the contact information (e.g., the IP address) of each VoIP user. Two types of SIP servers are deployed in NTP VoIP platform, including the call server developed by *Industrial Technology Research Institute* (ITRI) and the *SIP Express Router* (SER) developed by iptel. The ITRI call server is a commercial product and provides a convenient *Operation, Administration and Maintenance* (OAM) interface. The SER is an open-source server deployed on a Linux or a BSD system. In this paper, we utilize SER to develop a monitoring system and a conference system.
- **PSTN gateway** (Figure 1(5)) supports interworking between NTP VoIP platform (i.e., an IP network) and the *Public Switched Telephone Network* (PSTN), which allows VoIP phone users to reach other PSTN users. Three PSTN gateways have been deployed in NTP VoIP platform, including the Vontel gateway developed by ITRI, Cisco 2621 and Cisco 3700. The Vontel gateway and Cisco 3700 provide E1 interfaces and each E1 interface can support 30 concurrent users. The Cisco 2621 provides four *Foreign eXchange Office* (FXO) interfaces that support 4 concurrent users. The SIP server dispatches the calls to these gateways based on a load balancing mechanism.
- **SIP User Agent** (UA; Figure 1(4)(6)(7)(8)) is a hardware-based or a software-based SIP phone that provides several call functions such as dial, answer, reject, hold/unhold, and call transfer. In

[1] This work was sponsored in part by NICI IPv6 Project, NTP VoIP Project 95-2219-E-009-010, 94-2219-E-009-001, and ITRI/NCTU Joint Research Center.

NTP VoIP platform, the software-based SIP UA has been installed in several terminals including desktop computers, notebooks, PDAs, and Wireless LAN/Cellular dual-mode handsets. We have also integrated and tested the hardware-based SIP phones manufactured by BCM (WiFi), Polycom (video), Leadtek (video), Innomedia (video), Cisco, Pingtel, and Snom.

- **Monitoring system** (Figure 1(3)) is developed in this paper to provide *Lawful Interception* for the *Law Enforcement Agency*. The monitoring system provides the call detail records and the interception function. In NTP VoIP platform, this monitoring system is a plug-in system. The original SIP server does not require any modification. Moreover, the monitoring system can intercept the multimedia calls (e.g., a voice call or a video call) from all kinds of SIP UAs including WiFi phones and video phones.
- **Conference system** (Figure 1(2)) is developed to provide video/voice conference communication for a group of users. The conference system includes two servers: Conference Allocator and Conference server. Conference Allocator is responsible for allocating conference timeslots among different groups of users, and Conference Server is responsible for performing video/voice conference functions, such as stream mixing and multicasting. Based on the two servers, we propose a video/voice conference system called NTP video/voice conference system.

This paper is organized as follows. First, we elaborate the related work such as SIP header fields, *Session Description Protocol* (SDP) [2] fields and *Media Gateway Control Protocol* (MCGP) [5] commands in Section 2. Sections 3 and 4 present the monitoring system and the conference system, respectively. The conclusions are given in Section 5.

2 Related Work

2.1 SIP and SDP

SIP signaling protocol is used to create, modify and terminate a multimedia session. IETF RFC 3261 defines six types of SIP request messages including *REGISTER*, *INVITE*, *ACK*, *CANCEL*, *BYE* and *OPTIONS*. *REGISTER* message is used to register the contact information (e.g., its IP address) of a SIP UA to the SIP server. After registration, a SIP UA can send an *INVITE* message to the called UA to initiate a session. *ACK* message is used to confirm the final response of the *INVITE* message. *CANCEL* message and *BYE* message are used to terminate a pending session and an ongoing session, respectively. A SIP UA sends an *OPTIONS* message to query the capability of anther UA or a SIP server. Above SIP request messages are confirmed by SIP response messages such as *200 OK*, *100 Trying*, *180 Ringing*, *302 Move Temporarily*, and *404 Not Found*.

Figure 2 illustrates a basic SIP call example between UA1 and UA2 through a SIP server. In this example, UA1 is the calling party and UA2 is the called party. Assume that UA1 and UA2 register their IP address to the SIP server. The procedure is described as follows.

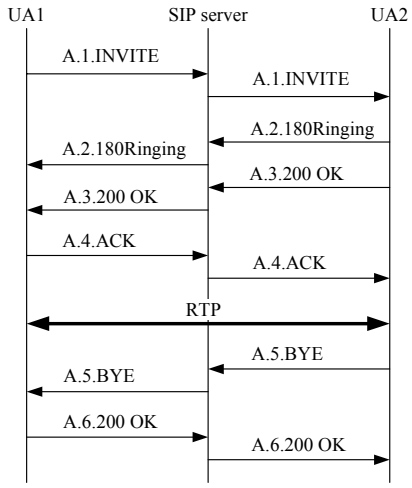

Figure 2. SIP Call Setup Procedure

Step A.1. UA1 sends an *INVITE* message, which carries the *Real-time Transport Protocol* (RTP) [3] information in the SDP fields, to UA2. Since UA1 does not know the contact IP address of UA2, it sends the *INVITE* message to the SIP server. The SIP server retrieves UA2's IP address and forwards the *INVITE* message to UA2.

Step A.2. Upon receipt of the *INVITE* message, UA2 plays a ringing tone and replies a *180 Ringing* to UA1. Based on the *Via* header field, the route of the *180 Ringing* message follows the reversed direction of the *INVITE* message. Therefore, this message is sent to the SIP server and then forwarded to the UA1. When UA1 receives the *180 Ringing* message, it plays the ring-back tone to notify the user of UA1 that UA2 is ringing.

Step A.3. When the called party picks up the call, UA2 sends a *200 OK* message to UA1. The *200 OK* message carries the RTP information of UA2. The *200 OK* message follows the same path as *180 Ringing* message to UA2.

Step A.4. Upon receipt of the *200 OK* message, UA1 sends an *ACK* message to UA2. At this point, the RTP session between UA1 and UA2 is established.

Steps A.5 and A.6. Any party of the RTP session can send a *BYE* message to terminate the ongoing RTP session. In this example, UA2 sends the *BYE* message to UA1. UA1 replies a *200 OK* message to UA2 to confirm that the RTP session is terminated.

The following is an example of the *INVITE* message that is received by the SIP server at Step A.1. We utilize this example to present the SIP header fields and the SDP fields that are used in this paper.
The SIP header fields:

```
INVITE sip:UA2@SIP_server SIP/2.0
Via: SIP/2.0/UDP UA1.work.com
From: Alan <sip:UA1@ SIP_server>
To: Bob <sip:UA2@ SIP_server>
Call-ID: 123456@ UA1.work.com
CSeq: 1 INVITE
Content-Length: 421
Content-Type: application/sdp
v=0
c=IN IP4 140.113.131.10
m=audio 9000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

- The *Request-URI* field (e.g., sip:UA2@SIP_server) indicates the *Uniform Resource Identifier* (URI) [4] of the receiver (UA2). A SIP request message is forwarded based on this field.
- The *To* field (e.g., Bob <sip:UA2@ SIP_server>) contains an optional display name of the transaction target (which could be a person or a system, i.e., Bob) followed by its SIP URI sip:UA2@ SIP_server.
- The *From* field (e.g., Alan <sip:UA1@ SIP_server>) contains an optional display name of the transaction initiator (Alan) followed by its SIP URI sip:UA1@ SIP_server.
- The *Via* field (e.g., SIP/2.0/UDP UA1.work.com) contains the version (SIP/2.0) and the transport protocol (UDP) followed by the IP address/port number of the immediate sender (UA1). Any intermediate server that forwards the SIP message adds a *Via* field with its address and port number. This field may also be expressed with a domain name (e.g., UA1.work.com).
- The *Content-Length* field counts the SIP message body in octets. A *Content-Length* 0 indicates that there is no message body.

The SDP fields:

- The *c* field (e.g., c=IN IP4 140.113.131.10) indicates the connection information for media (voice or data) session, which includes the network type (IN), address type (IP4), and connection address (which can be the originator's IP address 140.113.131.10). This field may also be expressed with a domain name.
- The *m* field (e.g., m=audio 9000 RTP/AVP 0) indicates the media, which contains the media type (audio), port (9000), protocol (RTP/AVP), and the codec number (i.e., 0 for u-law PCM).

2.2 MGCP Commands and Parameters

A controller referred to as a call agent uses MGCP [5] to instruct the *Media Gateway* (MG) to process the multimedia streams such as the RTP streams. In this paper, MGCP is used in the proposed VoIP monitoring system. The related MGCP commands are described as follows.

- The **CreateConnection** (CRCX) command is issued by the call agent to an MG and used to create a new connection.
- The **ModifyConnection** (MDCX) command is issued by the call agent to an MG and used to modify the parameters of a connection.

The parameters are described as follows.

- The **CallId** (C) parameter is the unique identifier for a call and composed of hexadecimal numbers.
- The **ConnectionId** (I) parameter is the unique identifier for a connection on an MGCP endpoint and composed of hexadecimal numbers.
- The **ConnectionMode** (M) parameter is used to configure the connection of an endpoint to receive-only, send-only or send-receive mode.

3 VoIP Monitoring System

For lawful interception in a SIP-based VoIP system, we propose a monitoring system (Figure 1(3)) that provides the *Call Detail Records* (CDRs) and the interception function. Figure 3 illustrates the major components of the monitoring system including a monitoring portal (Figure 3(4)), a monitoring server (Figure 3(2)), and a RTP proxy (Figure 3(3)).
To monitor a phone call, the *Law Enforcement Agency* (LEA) configures the monitoring system and requests the VoIP operator referred to as an *Internet Telephony Service Provider* (ITSP) to forward the SIP messages to the monitoring system. In the proposed monitoring system, an administrator (i.e., the LEA) can configure the information of a monitored user (e.g., a suspect), browse the CDRs and obtain the intercepted voice/video files on the monitoring portal through *Hyper Text Transfer*

Protocol (HTTP). The monitoring portal sends a SIP *Instant Message* (IM) to notify the administrator while receiving a voice/video file. To implement the monitoring server, we develop a monitoring module and embed this module into the IPtel SER [6].

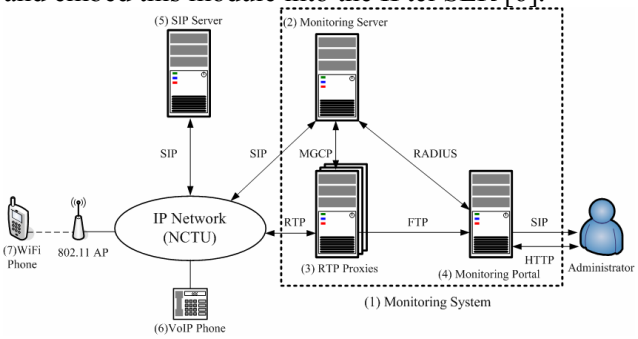

Figure 3. The VoIP Monitoring System

The monitoring module receives exams the incoming SIP messages based on the user's information retrieved from the monitoring portal, modifies the SDP *c* and *m* fields to redirect the monitored RTP streams to the RTP proxy and finally stores the CDRs to the monitoring portal. This monitoring module communicates with the monitoring portal through *Remote Authentication Dial-In User Service* (RADIUS) and communicates with the RTP proxy through MGCP. The RTP proxy records the incoming RTP streams and then forwards these streams to the destinations. Since the loading of a RTP proxy is larger than that of a monitoring server, the monitoring server dispatches the RTP streams to multiple RTP proxies for reducing the loading of a RTP proxy. Under this architecture, a monitored user does not easily detect the existence of a RTP proxy, but the intercepted voice/video files will be distributed among the multiple RTP proxies. To solve this problem, each RTP proxy uploads the intercepted files to the monitoring portal through *File Transfer Protocol* (FTP), and the administrator can retrieve all files from the monitoring portal.

We utilize a call setup example (Figure 4) to elaborate the operation of the monitoring system. In this example, the WiFi phone with the phone number 0944300001 is the calling party and binds on IP address 140.113.131.70. The VoIP phone with phone number 0944300002 is the called party and listens on IP address 140.113.131.90. The phone number 0944300002 is a monitored number. To simplify the description, we assume that the WiFi phone and the VoIP phone register to the monitoring server. The RTP proxy listens on the IP address 140.113.131.80.

Step B.1. The WiFi Phone sends a SIP *INVITE* message to the monitoring server, where the *Request URI* is 0944300002@SIP_server, the SDP *c* field contains IP address 140.113.131.70 and the *m* field contains port number 9000.

Step B.2. Upon receipt the *INVITE* message, the monitoring server checks the SIP *Request URI*, *From* and *To* header fields. Since the *Request URI* header field contains the monitored number 0944300002, the monitoring server issues an MGCP *CreateConnection* message to the RTP proxy to start the monitoring procedure. In this MGCP message, the *c* and *m* fields contain the wildcard '$' symbols that are used to request an IP address and a port number from the selected RTP proxy.

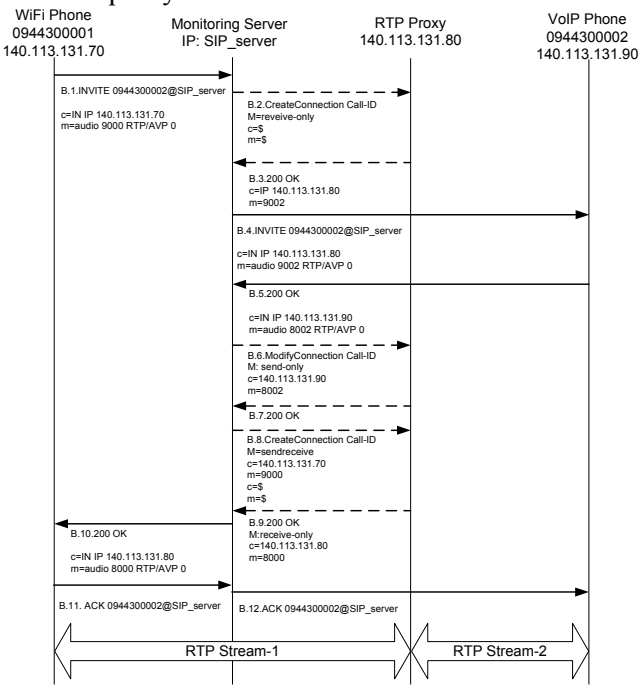

Figure 4. A Call Setup Example

Step B.3. The RTP proxy returns its IP address 140.113.131.80 and an available port number 9002 in an MGCP *200 OK* message.

Step B.4. Upon receipt of the *200 OK* message, the monitoring server modifies the IP address and the port number of the SDP *c* and *m* fields of the *INVITE* message that is received at Step B.2. The monitoring server modifies the host address (i.e., SIP_server) of the *Request URI* header field to 140.113.131.90 based on the registration information of the VoIP phone. Then the monitoring server forwards the modified *INVITE* message to the VoIP phone according to the *Request URI*.

Step B.5. When the VoIP phone receives the *INVITE* message, it obtains the RTP information from the SDP *c* and *m* fields. After the called party has picked up the call, the VoIP phone replies a SIP *200 OK* message to the WiFi phone. This message carries the IP address 140.113.131.90 and port number 8002 of the RTP connection in the SDP *c* and *m* fields. The *200 OK* message is sent back to the monitoring server according to

the top *Via* header field (SIP/2.0/UDP SIP_server:5060). Therefore, the route of the *200 OK* message is the reverse of that for the *INVITE* message.

Step B.6. Upon receipt of the *200 OK* message, the monitoring server obtains the RTP information of the VoIP phone from the SDP *c* and *m* fields (i.e., the IP address is 140.113.131.90 and the port number is 8002) and sends this information to the RTP proxy through MGCP *ModifyConnection* message.

Step B.7. The RTP proxy replies an MGCP *200 OK* message after it has built the RTP mapping for RTP Stream-2.

Step B.8. The monitoring server sends the WiFi phone's address 140.113.131.70 and port number 9000 to the RTP proxy and instructs the RTP proxy to build the mapping for RTP Stream-1.

Step B.9. After the mapping for RTP Stream-1 is built, the RTP proxy returns its IP address 140.113.131.80 and port number 8000 to the monitoring server through an MGCP *200 OK* message. Then the monitoring server modifies the SDP *c* and *m* fields of the SIP *200 OK* message based on the MGCP *200 OK* message.

Step B.10. The SIP *200 OK* message is sent back to the WiFi phone according to the top *Via* header field (SIP/2.0/UDP 140.113.131.70:5060).

Step B.11 and B.12. The *ACK* message is returned to the VoIP phone. The processing of the *ACK* message is similar to that of the INVITE message, and the details are omitted here.

After Step 12, the RTP connection is established between the WiFi phone and the VoIP phone through the RTP proxy assigned by the monitoring server. Therefore, the RTP proxy can intercept the voice of the monitored number.

Since the monitoring server processes all SIP messages of the monitored number, the monitoring server can obtain whole information including the calling party, the called party, the start time, the stop time and the duration. The monitoring server stores the CDR of a call into the monitoring portal after the call is complete.

4 Conference System

In the conference system, two servers (Conference Server and Conference Allocator) are provided to perform the conference session. Conference Server is used to mix the video/voice steams from different users and then multicasts the mixed stream to the group of users. In the NTP platform, the conference phone numbers, 0944022001~0944022100, are registered for Conference Server. Conference Allocator allocates the conference timeslots to different groups of users. As shown in Figure 5, we use an example to elaborate the execution flow of the video/voice conference system, which is executed in the following steps.

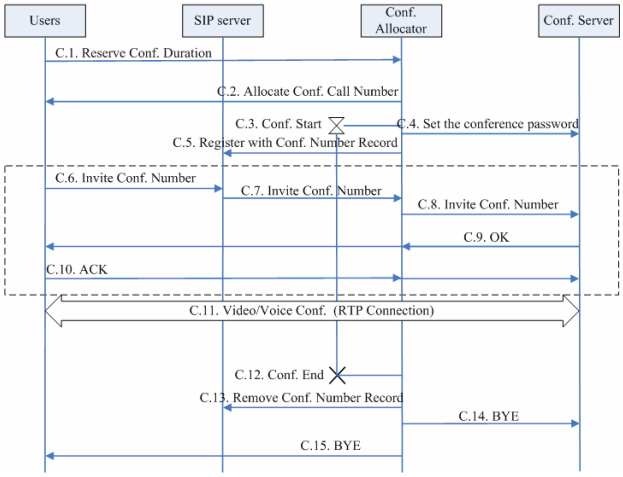

Figure 5. The Execution of the Conference System

Step C.1. One of the group users first reserves a conference timeslot (e.g., from 2:00pm to 3:00pm on 16 Feb. 2006) at Conference Allocator by using *Hypertext Transfer Protocol* (HTTP) request. In Conference Allocator, a *time table* is maintained to record the reservations of conference timeslots. Upon receipt of the reservation request, Conference Allocator checks whether the requested timeslot is available. If so, the requested timeslot is added to the time table. Otherwise, Conference Allocator rejects the reservation request.

Step C.2. When Conference Allocator accepts t`he reservation request, it chooses a conference phone number (e.g., 0944022001) by checking its time table and generates a password for this conference. Then Conference Allocator sends the conference phone number and the conference password to the requested user. The conference phone number is used to identify a specific conference, and the conference password is used to verify the authority for the access of Conference Server. Upon receipt of the phone number and password information, the user notifies other users participating in the conference about this information.

Step C.3. When the start-time of the conference timeslot is reached, Conference Allocator starts a *timer* that is used to hold the conference timeslot. For example, the timer is started at 2:00pm and expired at 3:00pm for this conference.

Step C.4. Conference Allocator sets the password

(generated at Step C2) for the conference phone number at Conference Server.

Step C.5. Conference Allocator registers the conference phone number to SIP server to add its contact information by using a SIP *REGISTER* message. This registration informs SIP server to route the SIP message (which invites the conference phone number) to Conference Allocator.

Step C.6. A user (involving this conference) sends the SIP *INVITE* message with the conference phone number to SIP server. Note that each group user must execute Steps C.6-C.10 individually.

Step C.7. When SIP server receives the *INVITE* message, it checks with the SIP registrar for the contact information of conference phone number. If the contact information is contained in the SIP registrar, the INVITE message is routed to Conference Allocator. Otherwise, this request is rejected.

Step C.8. Upon receipt of the *INVITE* message, Conference Allocator forwards this message to Conference Server. Conference Allocator also maintains a *conference table* to record the users who join the conference session.

Step C.9. Upon receipt of the first *INVITE* message for a conference phone number, Conference Server starts the video/voice conference session. If the other INVITE message is received, Conference Server adds the user to join the conference session. An *Interactive Voice Response* (IVR) stream is played to request the user providing the conference password (obtained at Step C.2 and set at Step C.4). If the password is correct, a SIP *200 OK* message is replied to the user through Conference Allocator.

Step C.10. Upon receipt of the *200 OK* message, the user sends an *ACK* message to Conference Server through Conference Allocator.

Step C.11. Upon receipt of the *ACK* message, the RTP connection between the user and Conference Server is established, and the video/voice conference is activated. Note that a user can join the video/voice conference between the start and the end of the conference timeslot, and Steps C.6-C.10 can be executed after Step C.11.

Step C.12. When the timer (started at Step C.3) is expired, Conference Allocator terminates the conference session by executing Steps C13-C15.

Step C.13. Conference Allocator removes the contact information of the conference phone number by sending a SIP *REGISTER* message to SIP server (i.e., the contact information is set to blank). Therefore, other *INVITE* messages will not be routed to Conference Allocator for this conference.

Steps C.14. Conference Allocator sends the SIP *BYE* message to Conference Server to terminate the conference session.

Step C.15. By querying the conference table (described at Step C7), Conference Allocator sends the *BYE* messages to the users involving this conference session.

After Step C15, the RTP connections between users and Conference Server are disconnected, and the conference session is finished.

5 Conclusion

This paper proposes a monitoring system for lawful interception and a conference system for audio and video conferences. The monitoring system provides call detail records and interception function. Through this system, the Lawful Enforcement Agency can intercept the VoIP calls of a suspect. On the other hand, through the conference system, a user can dynamically reserve the timeslot of a VoIP conference. We utilize the detail message flows to demonstrate the monitoring system and the conference system. The monitoring system won the top prize at the Taiwan Ministry of Education (MoE) Communication Contest 2005.

References:

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, IETF, June 2002.

[2] M. Handley and V. Jacobson. SDP: Session Description Protocol. RFC 2327, IETF, April 1998.

[3] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. IETF RFC-3550, July 2003.

[4] T. Berners-Lee, R. Fielding, L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. IETF RFC 2396. August 1998.

[5] F. Andreasen, B. Foster. Media Gateway Control Protocol (MGCP) Version 1.0. IETF RFC 3435. January 2003.

[6] iptel.org SIP Server: SIP Express Router. http://www.iptel.org/ser/