Experimental Design of a Laboratory for VoIP using SIP

Sergio A. Chacón, Driss Benhaddou, Deniz Gurkan

Engineering Technology Department
University of Houston

Abstract

A laboratory was designed to teach complex new technologies in computer networking to undergraduate and graduate students in the University of Houston's College of Technology. Our approach is based on a learning model that first focuses on knowledge comprehension and secondly on application, analysis and synthesis with emphasis in experimental methods that encourage constructive learning[1]. The experimental design of the laboratory for Voice over Internet Protocol (VoIP) using Session Initiation Protocol (SIP) is a result of this initiative.

Introduction

A rather new technology that has gained popularity, VoIP using the SIP involve various complex techniques and protocols which makes it difficult and lengthy during teaching practices. First, the principles for the conversion of analog voice to digital information must be explored together with encoding and compression algorithms. Second, the protocols involved in the transmission and networking should be explored. Finally, there is a wide variety of security issues involved.

For the delivery of the voice information, Real-Time Transport Protocol (RTP) is utilized with User Datagram Protocol (UDP). Segments are then passed to IP and the lower layers. In addition to the transport and network mechanisms, the IETF signaling protocol SIP, is used for managing call setup and registration of user agents, in most cases, a SIP proxy is installed to handle all of the signaling procedures employing the transport protocols UDP or Transport Control Protocol (TCP).

VoIP has to deal with many performance and security issues inherent in IP that create difficulties when trying to deliver voice information in real time over the Internet. Packetized voice must go through a series of routers and switches before it arrives to its destination that causes delay. Also, the integrity and confidentiality of the voice information must be guaranteed while using the same tools available in the Internet. Firewalls, Virtual Private Networks (VPN), Internet Protocol Security (IPSec) and Secure Socket Layer (SSL) are some of the tools available to secure voice.

The experimental design of the VoIP laboratory focuses on student participation during a learning process. The laboratory has been divided into four sections that intend to lead the student to learn by practicing. They are the Pre-Laboratory, the First and Second Procedures, and the Application sections.

Pre-Laboratory

The first section, the Pre-Laboratory, is research and simulation intensive and can actually be performed by the student before beginning the lab work. The Pre-Lab is initiated with the conceptualization of the project at hand through a series of definitions and problem solving strategies that lead to understanding how VoIP technology works, from the digitalization of the analog voice signal to how the RTP (RFC 3550) and SIP (RFC 3261) protocols enable voice communication over the Internet protocol.

The first part of the Pre-Lab involves a simulation in MatLab that demonstrate the digitalization of analog voice by sampling a voice signal. The MatLab program samples the voice signal at 8000 KHz (Nyquist theorem) with a bit depth of 8 bits or 256 voltage levels ($2^8=256$) to produce a 64Kbps bitstream. The sampled voice information is then encoded using Pulse Code Modulation µLaw (PCM µLaw), a logarithmic compression algorithm (G.711standard) that takes a 12-bit linear PCM sample and maps it into an 8-bit value[2]. The resulting audio stream is saved in a data file.

In the second part of the Pre-Lab, a C program is used to simulate the creation of voice packets. The student is required to do some research on the transport protocol RTP, determine the function of each RTP header components and explain why RTP needs UDP for transport of the voice packets. The program uses the saved audio file of the first part to produce payloads, goes through the encapsulation process by adhering the RTP, UDP and IP headers to these payloads, and finally attaches Ethernet headers and trailers (see Figure 1).

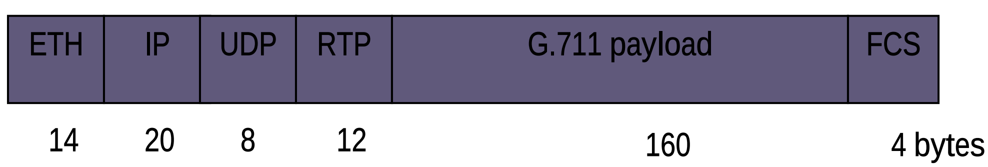

Figure 1. Voice frame

In the final part of the Pre-Lab, SIP is introduced as the signaling protocol. In this part, the student is asked to explain how SIP works. Basically, the signaling process with an overview of the Multipurpose Internet Mail Extensions (MIME) messages exchanged by the user agents should be covered as shown in Figure 2. In most cases, SIP messages are exchanged using UDP as the transport protocol.

After completing the Pre-Lab, the student will have an understanding of how voice information is converted from an analog form to a compressed digital form, how RTP and the Internet protocols facilitates the transport of voice information over the Internet network, and how SIP makes call setup possible.

```
                   SIP SRV
         .                       .
.                                         .
215.....................................221
|                  |                      |
|INVITE            |                      |
|----------------->|                      |
|   407 PROXY AUTH |                      |
|<-----------------|                      |
|ACK               |                      |
|----------------->|                      |
|INVITE            |                      |
|----------------->|INVITE                |
|       100 TRYING |--------------------->|
|<-----------------|          180 RINGING |
|      180 RINGING |<---------------------|
|<-----------------|               200 OK |
|           200 OK |<---------------------|
|<-----------------|                      |
|                 ACK                     |
|---------------------------------------->|
|           MEDIA SESSION                 |
|<=======================================>|
|                 BYE                     |
|<----------------------------------------|
|               200 OK                    |
|---------------------------------------->|
```

Figure 2. SIP signaling process

First Procedure

The First Procedure intends to reinforce what the student has learned in the previous Pre-Lab section. In this first part of this section, students are required to implement a basic VoIP setup, the Direct-Call setup, shown in Figure 3.

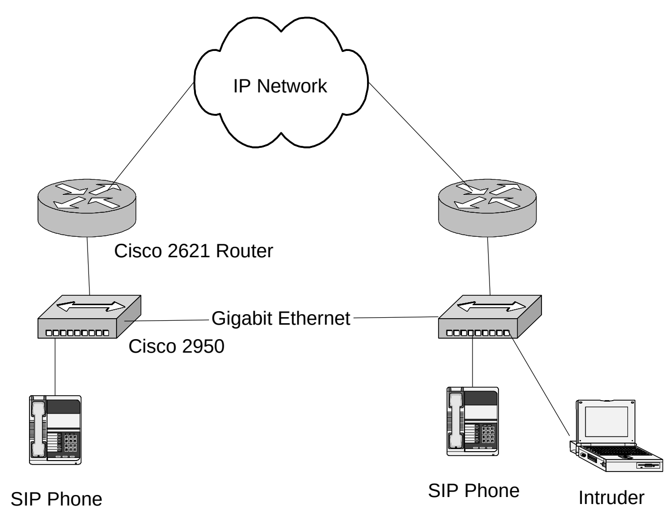

Figure 3. Direct-call setup

In this setup, two SIP phones are connected to Cisco switches placed at different locations. Both phones are in the same subnet. The student posing as an intruder and using the open-source network monitoring software Ethereal connects to one of the switches. One of the switch ports has been previously enabled for monitoring network traffic. A packet capture is initiated with Ethereal and a call is established. The result of the traffic monitoring shows captured SIP and RTP packets.

With the capture of SIP messages, the student can see the signaling process in action and study the MIME messages in detail such as the INVITE shown in Figure 4.

```
⊞ Frame 25 (921 bytes on wire, 921 bytes captured)
⊞ Ethernet II, Src: 00:0b:82:02:ca:2e, Dst: 00:50:8d:6e:63:33
⊞ Internet Protocol, Src Addr: 172.16.1.23 (172.16.1.23), Dst Addr: 172.16.1.29 (172.16.1.29)
⊟ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
    Source port: 5060 (5060)
    Destination port: 5060 (5060)
    Length: 887
    Checksum: 0x54ca (correct)
⊟ Session Initiation Protocol
  ⊟ Request-Line: INVITE sip:221@172.16.1.29;user=phone SIP/2.0
      Method: INVITE
      [Resent Packet: False]
  ⊟ Message Header
      Via: SIP/2.0/UDP 172.16.1.23;branch=z9hG4bK2a7ce123c9104946
    ⊟ From: <sip:215@172.16.1.29;user=phone>;tag=52666d33764cab19
        SIP from address: sip:215@172.16.1.29
        SIP tag: 52666d33764cab19
    ⊟ To: <sip:221@172.16.1.29;user=phone>
        SIP to address: sip:221@172.16.1.29
      Contact: <sip:215@172.16.1.23;user=phone>
      Call-ID: d5c8e8faa5e0eddc@172.16.1.23
      CSeq: 39864 INVITE
      User-Agent: Grandstream BT100 1.0.5.11
      Max-Forwards: 70
      Allow: INVITE,ACK,CANCEL,BYE,NOTIFY,REFER,OPTIONS,INFO,SUBSCRIBE
      Content-Type: application/sdp
      Content-Length: 392
  ⊟ Message body
    ⊟ Session Description Protocol
        Session Description Protocol Version (v): 0
      ⊞ Owner/Creator, Session Id (o): 215 8000 8000 IN IP4 172.16.1.23
        Session Name (s): SIP Call
      ⊞ Connection Information (c): IN IP4 172.16.1.23
      ⊞ Time Description, active time (t): 0 0
      ⊞ Media Description, name and address (m): audio 5004 RTP/AVP 0 8 4 18 2 15 99 9 101
      ⊞ Media Attribute (a): rtpmap:0 PCMU/8000
      ⊞ Media Attribute (a): rtpmap:8 PCMA/8000
      ⊞ Media Attribute (a): rtpmap:4 G723/8000
      ⊞ Media Attribute (a): rtpmap:18 G729/8000
      ⊞ Media Attribute (a): rtpmap:2 G726-32/8000
      ⊞ Media Attribute (a): rtpmap:15 G728/8000
      ⊞ Media Attribute (a): rtpmap:99 iLBC/8000
      ⊞ Media Attribute (a): fmtp:99 mode=20
      ⊞ Media Attribute (a): rtpmap:9 G722/8000
      ⊞ Media Attribute (a): ptime:20
      ⊞ Media Attribute (a): rtpmap:101 telephone-event/8000
      ⊞ Media Attribute (a): fmtp:101 0-11
```

Figure 4. INVITE message

After the phone call is established, SIP gives way to direct communication, media session (see Figure 2), between the two phones which uses RTP over UDP as the transport protocol. The student is asked to open one of the captured RTP packets and to identify the different components of the voice frame as shown in Figure 5. Some of the important components can be observed such as the source and destination IP addresses of the user agents in the IP header; source port, destination port and checksum in the UDP header; sequence number, timestamp, and payload in the RTP header. An analysis of the protocols through the captured packets leads the student to an understanding of their significance.

```
▿ Internet Protocol, Src Addr: 172.16.1.22 (172.16.1.22), Dst Addr: 172.16.1.23 (172.16.1.23)
    Version: 4
    Header length: 20 bytes
  ▹ Differentiated Services Field: 0x30 (DSCP 0x0c: Assured Forwarding 12; ECN: 0x00)
    Total Length: 200
    Identification: 0x0f89 (3977)
  ▿ Flags: 0x00
      0... = Reserved bit: Not set
      .0.. = Don't fragment: Not set
      ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 250
    Protocol: UDP (0x11)
    Header checksum: 0x561e (correct)
    Source: 172.16.1.22 (172.16.1.22)
    Destination: 172.16.1.23 (172.16.1.23)
▿ User Datagram Protocol, Src Port: 5004 (5004), Dst Port: 5004 (5004)
    Source port: 5004 (5004)
    Destination port: 5004 (5004)
    Length: 180
    Checksum: 0x234b (correct)
▿ Real-Time Transport Protocol
  ▿ [Stream setup by SDP (frame 29)]
      [Setup frame: 29]
      [Setup Method: SDP]
    10.. .... = Version: RFC 1889 Version (2)
    ..0. .... = Padding: False
    ...0 .... = Extension: False
    .... 0000 = Contributing source identifiers count: 0
    0... .... = Marker: False
    .000 0000 = Payload type: ITU-T G.711 PCMU (0)
    Sequence number: 3636
    Timestamp: 1583885594
    Synchronization Source identifier: 1894989723
    Payload: 6B6B6B6B6B6B6B6B6B6B6B6B6B6A6A6B6A6B6B6B6B6B6B6B...
```

Figure 5. Captured voice packet

In the second part of this section, the challenges of transmitting voice over the Internet protocol are explored[3]. Through a series of questions, the students are led to discover some of performance and security issues facing VoIP. For instance,

- voice packets are transmitted in real time through routers and switches in the network;
- a minimum of delay is needed to deliver good quality voice; and
- in order to accomplish timely delivery of voice at the destination, voice packets need to be buffered and reassembled using the sequence number in the RTP header.

Delay and jitter, the variance in delay, for the captured conversation can be observed in Ethereal using the Stream Statistics command.

In addition, some problems in VoIP security are presented such as eavesdropping, denial of service, and tampering. To demonstrate eavesdropping, the student is asked to produce an audio file from the captured payload using the Save Payload feature in Ethereal. This audio file can then be replayed with any media player.

At the end of this section, the student will have reinforced the acquired knowledge from the Pre-Lab and gained working experience on a basic VoIP configuration.

Second Procedure

Normally, a two-phone setup is seldom encountered. In most cases, thousands of phones need to be connected and hundreds of calls set up at the same time. Therefore, a SIP proxy is needed to register user agents and manage their calls. The student will be asked in the first part of this section to configure and install a SIP proxy as shown in Figure 6.

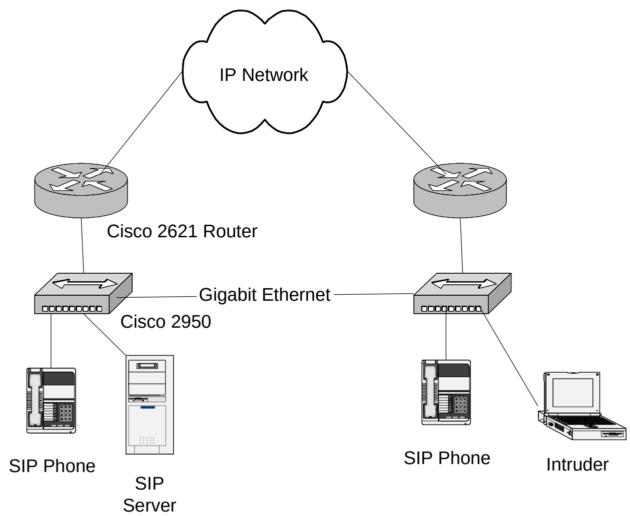

Figure 6. SIP proxy procedure

The SIP proxy requires two software programs: Linux as the operating system and open-source Asterisk[4]. Asterisk is a Private Branch Exchange (PBX) program that allows the proxy to be configured for SIP. Two files need to be configured with Asterisk: the sip.conf and the extensions.conf. The student will be given the information necessary to perform a basic configuration. Since the proxy is publicly accessible, further steps need to be taken to strengthen the proxy against possible attacks. From the performance point of view, the addition of the proxy does not add delay since, once the call setup is completed, the SIP proxy is relieved of its call managing duties allowing both phones to communicate directly.

In addition, the SIP phones need to be reconfigured to operate with the SIP proxy. In other words, the phones are registered with the proxy with the sip.conf file and how the SIP proxy is to manage calls between the phones with the extensions.conf file. Each phone is also given usernames and passwords for authentication purposes. Figure 2 actually shows how SIP handles the signaling with the proxy present in the network.

The student must then run another packet capture running Ethereal. From this capture it will be clear that the proxy acts as an intermediary and that registered and only properly authorized phones are given access to the VoIP system. When an INVITE message is received by the proxy, the proxy will first request authentication by sending a 407 PROXY AUTH message to the caller (see Figure 2). Despite added security, the student will see that the voice information is still vulnerable to eavesdropping from an intruder.

In the second part of this section, the student must write the conclusions. He/She is required to write a report that must include what is observed while performing the simulations and investigate possible solutions to some of the performance or security issues studied in a case study. An example of a case study is provided in the next section.

Application

There are several technologies used today to protect voice information from eavesdropping. Among them, Secure Socket Layer (SSL) and Virtual Private Network over Internet Protocol Security (VPN IPSec) are two of the most common solutions found[5]. The application presented here is one approach to the security of the voice information with VPN IPSec.

Solution
The proposed solution consists of placing two VPN-capable firewalls in the network between the switch and the user agents as shown in Figure 7. The firewalls use IPSec tunneling to encrypt the voice information. This solution is preferred over using VPN either in the border routers because it offers better protection against eavesdropping by an intruder inside the network. In the IEEE Africon 2004 conference, Aire[6] presented a similar solution; however, he placed the firewalls between the routers and the switches and the SIP proxy was placed inside one of the private networks while in this proposal the SIP proxy is placed in a Demilitarized Zone (DMZ) and is publicly accessible.

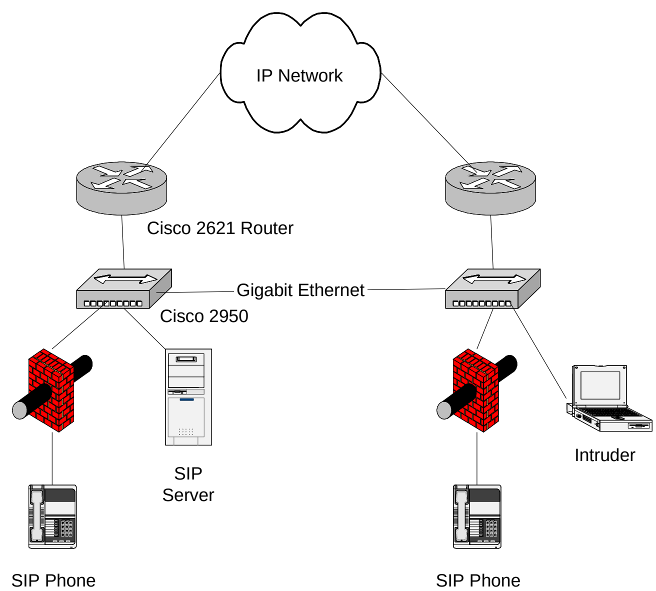

Figure 7. Application

Implementation
Two firewalls using Linux and open-source IPChains[7] and FreeS/WAN[8] with two NIC cards are installed. The main FreeS/WAN configuration file is /etc/ipsec.conf, shown in Figure 8, that uses /etc/ipsec.secrets which contains a public-private key pair to identify the host[9]. VPN over IPSec is configured specifying the interface used, processing options, and connections. The SIP phones are reconfigured with new private IP addresses

auto-start

corresponding to the left and right phone private networks and Network Address Translation (NAT) enabled.

```
config setup
        interfaces=%defaultroute
        klipsdebug=none
        plutobug=none
        plutoload=%search
        plutostart=%search
        uniqueids=yes
        nat_traversal=yes
        virtual_private=%v4:10.0.0.0/8,%v4172.16.0.0/12,v4:192.168.0.0/16,
        v4:!20.0.0.0/255.255.255.0,v4:!172.16.1.16/255.255.255.240,v4:!10.
        0.0.0/255.255.255.0

conn %default
        keyentries=0
        disablearrivalcheck=no

conn voip
        left=172.16.1.20
        leftnexthop=%defaultroute
        leftsubnet=20.0.0.0/255.255.255.0
        right=172.16.1.21|
        rightsubnet=10.0.0.0/255.255.255.0
        rightnexthop=%defaultroute

        ike==aes128-shamodp1536, aes128-shamodp1024, aes128-md5-modp1536,
        aes128-md5-modp1024,3des-sha-modp1536, 3des-sha-modp1024,
        3des-md5-modp1536,3des-md5-modp1024
        esp=aes128-sha1,aes128-md5,3des-sha1,3des-md5
        ikelifetime=1h
        keylife=8h
        dpddelay=30
        dpdtimeout=120
        dpdaction=hold
        authby=secret
        auto-start
```

Figure 8. Ipsec.conf

Demonstration

After the network has been implemented and tested, a packet capture was performed. The results of the capture are shown in Figure 9.

```
 74 33.608168 172.16.1.21    172.16.1.29    SIP/SD Request: INVITE sip:215@172.16.1.29;user=phone, with sessio
 75 33.609036 172.16.1.29    Broadcast      ARP    Who has 172.16.1.30?  Tell 172.16.1.29
 76 33.609048 172.16.1.29    Broadcast      ARP    Who has 172.16.1.30?  Tell 172.16.1.29
 77 33.609744 172.16.1.30    172.16.1.29    ARP    172.16.1.30 is at 00:00:0c:07:ac:03
 78 33.609747 172.16.1.29    10.0.0.2       SIP    Status: 407 Proxy Authentication Required
 79 33.615403 172.16.1.21    172.16.1.29    SIP    Request: ACK sip:215@172.16.1.29;user=phone
 80 33.618918 172.16.1.21    172.16.1.29    SIP/SD Request: INVITE sip:215@172.16.1.29;user=phone, with sessio
 81 33.619490 172.16.1.29    10.0.0.2       SIP    Status: 100 Trying
 82 33.620425 172.16.1.29    20.0.0.2       SIP/SD Request: INVITE sip:215@20.0.0.2;user=phone, with session de
 83 33.620780 172.16.1.29    20.0.0.2       SIP/SD Request: INVITE sip:215@20.0.0.2;user=phone, with session de
 84 33.625327 172.16.1.20    172.16.1.29    SIP    Status: 100 trying
 85 33.625331 172.16.1.20    172.16.1.29    SIP    Status: 100 trying
 86 33.626451 172.16.1.20    172.16.1.29    SIP    Status: 180 ringing
 87 33.626455 172.16.1.20    172.16.1.29    SIP    Status: 180 ringing
 88 33.626816 172.16.1.29    10.0.0.2       SIP    Status: 180 Ringing
 95 36.560664 172.16.1.21    172.16.1.20    ARP    Who has 172.16.1.20?  Tell 172.16.1.21
 96 36.560668 172.16.1.20    172.16.1.21    ARP    172.16.1.20 is at 00:e0:29:2d:15:f3
 97 37.374128 172.16.1.21    172.16.1.29    UDP    Source port: 5060  Destination port: 5060
118 48.422048 172.16.1.20    172.16.1.29    SIP/SD Status: 200 OK, with session description
119 48.422058 172.16.1.20    172.16.1.29    SIP/SD Status: 200 OK, with session description
120 48.422457 172.16.1.29    20.0.0.2       SIP    Request: ACK sip:215@20.0.0.2;user=phone
121 48.422718 172.16.1.29    20.0.0.2       SIP    Request: ACK sip:215@20.0.0.2;user=phone
122 48.423104 172.16.1.29    10.0.0.2       SIP/SD Status: 200 OK, with session description
123 48.423700 172.16.1.29    20.0.0.2       SIP/SD Request: INVITE sip:215@20.0.0.2, with session description
124 48.424051 172.16.1.29    20.0.0.2       SIP/SD Request: INVITE sip:215@20.0.0.2, with session description
125 48.429554 172.16.1.21    172.16.1.29    SIP    Request: ACK sip:215@172.16.1.29
126 48.429772 172.16.1.20    172.16.1.29    SIP/SD Status: 200 OK, with session description
127 48.429776 172.16.1.20    172.16.1.29    SIP/SD Status: 200 OK, with session description
128 48.429984 172.16.1.29    10.0.0.2       SIP/SD Request: INVITE sip:221@10.0.0.2, with session description
129 48.430245 172.16.1.29    20.0.0.2       SIP    Request: ACK sip:215@20.0.0.2
130 48.430496 172.16.1.29    20.0.0.2       SIP    Request: ACK sip:215@20.0.0.2
131 48.435602 172.16.1.21    172.16.1.29    SIP/SD Status: 200 OK, with session description
132 48.437849 172.16.1.29    10.0.0.2       SIP    Request: ACK sip:221@10.0.0.2
133 48.441283 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
134 48.443240 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
135 48.460517 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
136 48.462780 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
137 48.480503 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
138 48.482770 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
139 48.500497 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
140 48.502780 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
141 48.520507 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
142 48.522758 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
143 48.540479 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
144 48.542758 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
145 48.560476 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
146 48.562746 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
147 48.580470 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
148 48.582746 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
149 48.600457 172.16.1.20    172.16.1.21    ESP    ESP (SPI=0x37725695)
150 48.602746 172.16.1.21    172.16.1.20    ESP    ESP (SPI=0x971a4fa3)
```

Figure 9. Packet capture for demonstration

Summary and Conclusions

In summary, we have described the experimental design of the laboratory for Voice over Internet Protocol using SIP. The orientation of the experimental design given to the laboratory is based on initially giving the student a theoretical background while reinforcing this knowledge with a series of procedures that require application of concepts, analysis of the use of the technology and its consequences leading to problem solving. The design will be implemented in the University of Houston's College of Technology for the instruction of other complex technologies. This laboratory is being tested with undergraduate and graduate students in the College of Technology.

References

1. Haslam, E.L., 1997,"A Learning Model That Develops Students' Active Learning and Reflective Practices", *IEEE Frontiers in Education Conference*, Vol. 1, pp. 116-120.
2. http://en.wikipedia.org/wiki/G.711
3. Walsh, Thomas and Kuhn, Richard (2005). *Challenges in securing voice over IP*. IEEE Security & Privacy Magazine, Volume 3, Issue 3, 44-49
4. http://www.asterisk.org/
5. Kolesnikov, O., 2002,"Building Linux Virtual Private Networks (VPNs)", *New Riders*, pp 45-46.
6. Aire, F.T., Maharaj, B.T. and Linde, L.P. (2004). *Implementation considerations in a SIP-based secure voice over IP network.* Africon, 2004, 7th Africon Conference in Africa, Volume 1, 167-172
7. http://redhat.com
8. http://rpms.steamballoon.com/freeswan
9. McCarty, B., 2003,"Red Hat Linux Firewalls", *Red Hat Press*, pp 431-432.

SERGIO A. CHACON

Sergio Chacón serves as a Lab Manager in the Engineering Department of the University of Houston, Texas. His research interests include data communications technology, telecommunications, multimedia networking, and information systems security.

DRISS BENHADDOU

Dr. Benhaddou is an Assistant Professor at the Engineering Technology Department, University of Houston. His area of expertise includes optical networking, switching system design, routing protocols in optical networks, embedded systems, and network security.

DENIZ GURKAN

Dr. Deniz Gurkan joined the faculty of Department of Engineering Technology after receiving her PhD degree from the University of Southern California in May 2004. Her research interests are in optical fiber communications, biomedical sensors, sensor networks, and computer networking.