Open Source VoIP on Debian

FOSDEM 2007
Brussels, Belgium

Daniel Pocock
B.CompSc(Melbourne)
www.readytechnology.co.uk

Overview

- User expectations
- How it works
- Survey of available software
- Overview of reSIProcate
- sipdial example with reSIProcate
- Security analysis – SIP REFER
- Conclusions

User expectations and behaviour

- Convenient – e.g. address book dialing on mobile
- Free – e.g. Skype, MSN
- `Never' miss a call - mobile, any time, anywhere (library, restaurant, restroom)
- As developers, we must seek to deliver solutions that meet the needs and expectations of the user

How it works - overview

- Audio and video streams transmitted using RTP, which is encapsulated in UDP
- Call setup and related control activities executed with SIP (or H.323 or MGCP)
- We will focus exclusively on SIP, it is easy to understand, widespread and best positioned to challenge Skype, MSN, etc

How it works – audio streams

- 8kHz, 16bit unsigned linear audio samples are typical, just like ISDN
- Samples are compressed using a codec
- Frames of compressed data are embedded in RTP packets
- RTP packets embedded in UDP/IP and transmitted across network

How it works – codec comparison

Codec	Input	Frames/second	Frame size (bytes)	Bitrate (raw, kbps)	Bitrate (with headers, kbps) Assume 1 frame per packet	Bitrate (with headers, kbps) Assume 2 frames per packet	Notes
G.711U/A	16bit, 8kHz unsigned	100	80	64	80	72	Same as ISDN
G.729	16bit, 8kHz unsigned	100	10	8	24	14.4	Patented, good quality, low bandwidth
G.723.1	16bit, 8kHz unsigned	33 1/3	24 20	6.3 5.3	11.7 10.7	9.1 8.0	Patented, robotic sound Patented, more robotic
iLBC	16bit, 8kHz unsigned	33 1/3	50	13.3	18.7	17.3	Patent free, more CPU than G.729, less support in phones
GSM	16bit, 8kHz unsigned	50	32.5	13	21.2	17	More widely supported than iLBC, but not as good under packet loss

T.38 fax `codec'

Used for sending faxes. Not quite like other codecs, it deals with transmission of compressed image data rather than audio data, and it uses it's own transport mechanism rather than RTP in UDP. About 40kbps needed in one direction only.

How it works - SIP

- SIP = IETF Session Initiation Protocol
- Provides a means of starting and stopping calls
- Transmits information about caller (CLI), callee (e.g. phone number or username)
- Uses headers similar to HTTP and SMTP, embedded in UDP packets on port 5060. Therefore, easy for us to use our existing skills.

How it works - SIP

- SIP defines these entities:
  - User Agent (UA) – client or server
  - Proxy – stateless
  - Proxy - stateful
  - Registration Server
- Some applications/devices implement a combination of the above entities
- A proxy can only `relay' SIP packets, while a UA or Registration server can create new requests and responses

How it works - SIP

```
                                              bob
                                            +----+
                                            | UA |
                                            |    |
                                            +----+
                                               |
                                               |3)INVITE
                                               |   carol@chicago.com
      chicago.com        +--------+            V
      +---------+ 2)Store|Location|4)Query +-----+
      |Registrar|=======>| Service|<=======|Proxy|sip.chicago.com
      +---------+        +--------+=======>+-----+
            A                      5)Resp      |
            |                                  |
            |                                  |
  1)REGISTER|                                  |
            |                                  |
         +----+                                |
         | UA |<-------------------------------+
cube2214a|    |                            6)INVITE
         +----+                    carol@cube2214a.chicago.com
          carol

                   Figure 2: REGISTER example
```

Diagram courtesy of RFC 3261

How it works - SIP

- URIs are used, e.g. sip:daniel@lvdx.com, sip:442071357000@lvdx.com
- Each UDP message is a `request' or a `response', much like HTTP
- Typical requests are INVITE, REFER, BYE, MESSAGE, REGISTER
- Typical responses are 180 Ringing, 200 OK, 486 Busy, 503 Service Unavailable
- Digest auth scheme similar to HTTP auth

How it works - SIP

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142    (Alice's SDP not shown)

SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131    (Bob's SDP not shown)
```

SIP Dialog identification

The `From tag', `To tag' and `Call ID' form a tuple (the `Dialog ID') which uniquely identifies a `dialog'.

From tag: 1928301774
To tag: 1928301774
Call ID: a84b4c76e66710@pc33.atlanta.com

The SIP `dialog' is similar to a HTTP `Session', the `Dialog ID' is similar to a unique session cookie.

Survey of available software

Category	Name	Language	License	Debian packages	Description
PBX/Proxy/Server	repro	C++	Vovida (like BSD)		Highly scalable and extensible proxy for SIP, part of reSIProcate project
PBX/Proxy/Server	SER OpenSER	C	GPL or private GPL	Y Y	Highly scalable and configurable proxy for SIP OpenSER extends upon SER
PBX/Proxy/Server	Asterisk	C	GPL or private	Y	Popular PBX, acts as a UA rather than a SIP proxy, so it has less scalability but more features.
PBX/Proxy/Server	OpenPBX	C	GPL	Y	Spin off of Asterisk, by developers who believe less dependence on Digium is important
PBX/Proxy/Server	YATE	C++	GPL	Y	Extensible telephony architecture
Softphone	ekiga	C	GPL	Y	Softphone with V4L2 webcam support
Softphone	GnomeMeeting	C	GPL	Y	Softphone
Softphone	linphone	C	GPL	Y	Softphone
Softphone	kphone	C	GPL	Y	Softphone
Softphone	wengophone	C	GPL	Y	Softphone
Softphone	SIP Communicator	Java	GPL		Java softphone - multi-platform
Library	reSIProcate	C++	Vovida		Very thorough C++ implementation of SIP, multi-platform
Library	exosip	C	GPL	Y	Implements SIP
Library	ccRTP	C++	GPL	Y	Provides classes for handling RTP streams

Overview of reSIProcate

- rutil/* - provides basic classes across all platforms, e.g. Data, Thread
- resip/stack/* - for parsing SIP messages and SDP, typical classes are SipMessage, SdpContents, Uri, NameAddr
- resip/dum/* - for managing SIP dialogs from start to finish, typical classes are DialogUsageManager, InviteSessionHandler

Using DialogUsageManager

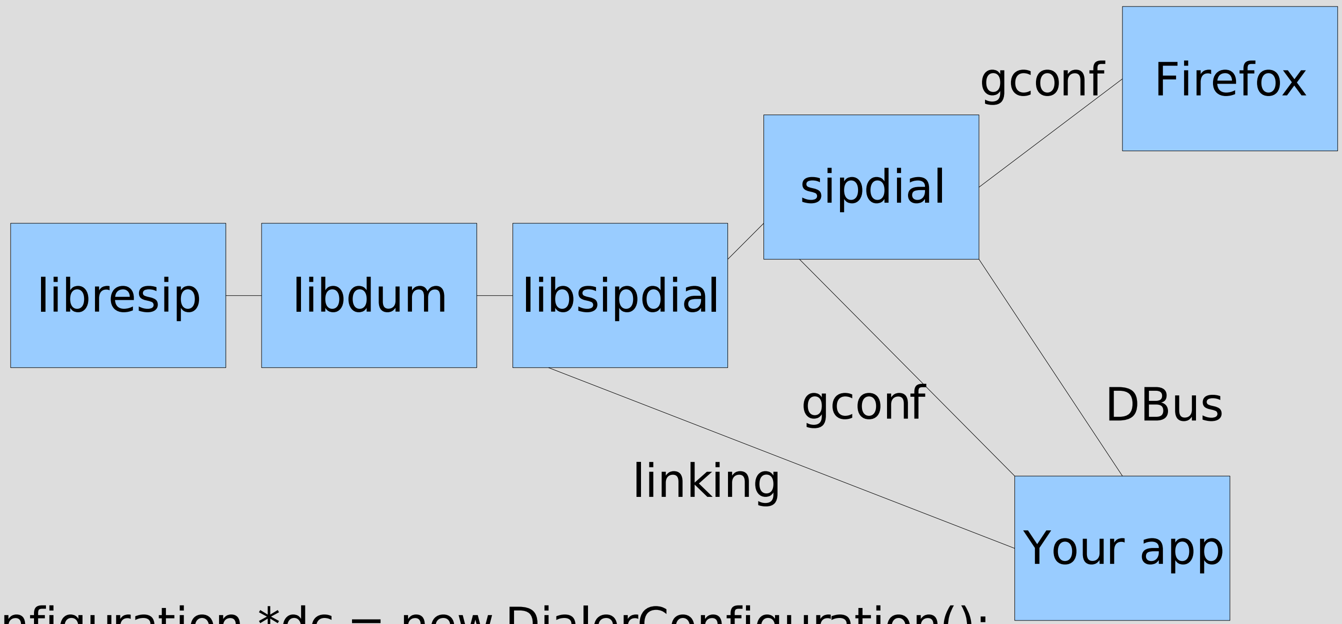
- Create subclass of InviteSessionHandler
- Create subclass of ServerAuthManager if needed
- Create instance of SipStack and DialogUsageManager
- Loop, calling SipStack::process() and DialogUsageManager::process()
- Respond to events inside MyInviteSessionHandler

Using DialogUsageManager

reSIProcate will call methods on MyInviteSessionHandler to tell us when important things are happening.

```
class MyInviteSessionHandler : public resip::InviteSessionHandler {

  // ...


  virtual void onFailure(resip::ClientInviteSessionHandle cis, const resip::SipMessage& msg);

  virtual void onConnected(resip::ClientInviteSessionHandle, const resip::SipMessage& msg);

  virtual void onReferAccepted(resip::InviteSessionHandle, resip::ClientSubscriptionHandle, const resip::SipMessage&
msg);

  virtual void onReferRejected(resip::InviteSessionHandle, const resip::SipMessage& msg);

  virtual void onNewSession(resip::ClientInviteSessionHandle cis, resip::InviteSession::OfferAnswerType oat, const
resip::SipMessage& msg);

  virtual void onTerminated(resip::InviteSessionHandle is, resip::InviteSessionHandler::TerminatedReason reason,
const resip::SipMessage* msg);

  // ...

};
```

An example - sipdial




```
DialerConfiguration *dc = new DialerConfiguration();
ifstream in(config_filename);
dc->loadStream(in);

DialInstance di(*dc, Uri("tel:+442071357000"));
di.execute();
```

An example - demo

- sipdial works with Linksys, Cisco and Polycom, probably others too
- Linksys has donated SPA-941 and other devices to demo and give away at FOSDEM

Security analysis – SIP REFER

- Can be sent mid-dialog, most phones will use their credentials when following REFER
- Who is at fault?
  - sipdial – ask for confirmation?
  - proxy – filter `nasty' REFERs?
  - SIP protocol – should the sender of REFER be responsible for call charges somehow?
  - handset – confirm before following REFER?

Conclusions

- Many apps will soon integrate with telephony and SIP
- Traditional telecommunications will be eroded by SIP. This is already happening in both the enterprise markets and the consumer markets.
- This is the ideal time for OPEN SOURCE and OPEN STANDARDS to establish themselves as the dominant paradigm.

Where to next?

- For free phone numbers, SIP services and links: www.opentelecoms.org
- For reSIProcate: www.reSIProcate.org
- For Linksys resellers: www.linksys.com