Rhizomatica

SIP Protocol Introduction

SIP Introduction

- SIP (Session Initialization Protocol) is an application-layer control protocol used to create, manage, terminate session in an IP based network. A simple session could be a simple two-way voice call or it could be a multimedia conference session.

- SIP is designed as a peer-to-peer protocol. P2P denote a network on which a node can act as server for others.

- SIP works between the session and application layers of the OSI model. This means that it can work in and between IPv4 and IPv6 model.

- SIP is a text based protocol it can be used with UDP or TCP. Can be also implemented with TLS to add encryption.

SIP Introduction

- To keep SIP as flexible as possible most of SIP's message and header syntax is derived from HTTP/1.1 specification
- SIP uses request/response transactions.
- SIP can work in a framework with other protocols like SOAP HTTP XML SDP etc..
- SDP (Session Description Protocol) is used to describe multimedia sessions
- RTP (Real-Time Transport Protocol) provides end-to-end delivery services for data (voice or video)
- SIP latest RFC is RFC 3261. SDP RFC 4566. RTP RFC 3550

SIP Functions

What does SIP do?

- Setup, establishing, tying together, and tear-down/terminating of multimedia sessions. SIP does this by providing 5 different functions.
- SIP is responsible for user location, user availability, user capabilities, session setup, and session management.

User Location

- SIP determines user locations by a registration process. When a soft-phone is activated on a laptop, it sends out a registration to the SIP server announcing availability to the communications network. Voice over-IP (VoIP) phones, cellular phones, or even complete teleconferencing systems can be registered as well. Depending on the registration point chosen, there may be several different locations registered simultaneously.

SIP Functions

User Availability

- User availability is simply a method of determining whether or not a user would be willing to answer a request to communicate. If you call and no one answers, SIP determines that a user is not available. A user can have several locations registered, but might only accept incoming communications on one device. If that is not answered, it can be transfer to another device, or transfers the call to another application, such as voicemail.

User Capabilities

- With all the various different methods and standards of multimedia communications, something is needed to check for compatibility between the communications and the users' capabilities. This function also determines which encryption/decryption methods a user can support.

SIP Functions

Session Setup

- SIP establishes the session parameter for both ends of the communications. more specifically, where one person calls and the other answers. SIP provides the means to setup and/or establish communications.

Session Management

- This function provides the greatest amount of user awe. Provided a device is capable, a user could transfer from one device to another, such as from an IP-based phone to a laptop, without causing a noticeable impact. SIP re-evaluates and modifies the communications streams to return the voice quality. With SIP session management, a user can also change a session by making it a conference call, changing a telephone call to a video conference. And finally, SIP terminates the communications. Session management

SIP Components

- Entity interacting in a SIP network are called UA (User Agents). User Agents may operate as:

- **UAC** (User Agent Client) it generates requests and send those to servers
- **UAS** (User Agent Server) it gets requests, processes those and generate responses.

- There is another entity called **B2BUA** (Back to Back User Agent) that act as both UAC and UAS.

SIP Components

Clients

- Client are usually the softphone or application

Servers

- Servers are in general part of the network. Servers can be of several types:

  Proxy Server, Registrar, Location Server, Redirect Server

SIP Components

- **Proxy Server**

  Most common type of server in a SIP enviroment. When a request is generated the exact address of the recipient is not know in advance. So the client sends the request to the Proxy Server. The server on behalf of the client forward the request to another proxy.

- **Registrar**

  Registrar server is to register the location of the client. One of the function of the Proxy Server is to detect the location of a user in a network. How do they know the location? By looking up on a registrar server. Users from time to time refreshes their registration

- **Location Server**

  Contains the addresses registered to the Registrar. Addresses for example can be saved on a database

- **Redirect Server**

  A redirect server redirects the request back to the client indicating the client to try with a different route to get to the recipient

SIP Basic Messages

- **INVITE**: Invites a user to a call
- **ACK**: Acknowledgement is used to facilitate reliable message exchange for INVITEs.
- **CANCEL**: Terminates a request. For example it's used when a client make a call and then changes it's decision to call the recipient
- **REGISTER**: Register a user current location
- **INFO**: Used for mid-session signaling
- **MESSAGE**: To send an IM message.
- **OPTIONS**: To get information about a server capabilities
- **UPDATE**: To update a current session

SIP Response Types

- For each request SIP UAS replies with a so called Status-Code.
- The first digit of a Status-Code defines the category of response. So any response between 100 and 199 is termed as a "1xx" response and so is done for any other type. SIP/2.0 allows six types of response. They are similar to those of HTTP.

SIP Response Types

- **1xx: Provisional** - request received, continuing to process the request
- **2xx: Success** - the action was successfully received, understood, and accepted
- **3xx: Redirection** - further action needs to be taken in order to complete the request
- **4xx: Client Error** - the request contains bad syntax or cannot be fulfilled at this server
- **5xx: Server Error** - the server failed to fulfill an apparently valid request
- **6xx: Global Failure** - the request cannot be fulfilled at any server

SIP Basic Call Flow

```
Alice                        Bob
     |                        |
     |       INVITE F1        |
     |----------------------->|
     |    180 Ringing F2      |
     |<-----------------------|
     |                        |
     |       200 OK F3        |
     |<-----------------------|
     |         ACK F4         |
     |----------------------->|
     |   Both Way RTP Media   |
     |<======================>|
     |                        |
     |         BYE F5         |
     |<-----------------------|
     |       200 OK F6        |
     |----------------------->|
     |                        |
```

SIP Basic Call Flow

```
F1 INVITE Alice -> Bob

   INVITE sip:bob@biloxi.example.com SIP/2.0
   Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
   Max-Forwards: 70
   From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
   To: Bob <sip:bob@biloxi.example.com>
   Call-ID: 3848276298220188511@atlanta.example.com
   CSeq: 1 INVITE
   Contact: sip:alice@client.atlanta.example.com;transport=tcp
   Content-Type: application/sdp
   Content-Length: 151

   v=0
   o=alice 2890844526 2890844526 IN IP4 client.atlanta.example.com
   s=-
   c=IN IP4 192.0.2.101
   t=0 0
   m=audio 49172 RTP/AVP 0
   a=rtpmap:0 PCMU/8000
```

SDP Session Description Protocol

Defines how the actual voice stream will be Established.
- Codecs offer
- IP address and port where to send
  The RTP stream

Tells Bob where to send the RTP stream.

```
 F2 180 Ringing Bob -> Alice

   SIP/2.0 180 Ringing
   Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101
   From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
   To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
   Call-ID: 3848276298220188511@atlanta.example.com
   CSeq: 1 INVITE
   Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
```

SIP Basic Call Flow

```
F3 200 OK Bob -> Alice

   SIP/2.0 200 OK
   Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bf9;received=192.0.2.101
   From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
   To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
   Call-ID: 3848276298220188511@atlanta.example.com
   CSeq: 1 INVITE
   Contact: <sip:bob@client.biloxi.example.com;transport=tcp>
   Content-Type: application/sdp
   Content-Length: 147
```

```
   v=0
   o=bob 2890844527 2890844527 IN IP4 client.biloxi.example.com
   s=-
   c=IN IP4 192.0.2.201
   t=0 0
   m=audio 3456 RTP/AVP 0
   a=rtpmap:0 PCMU/8000
```

SDP Session Description Protocol

Tells Alice where to send the RTP stream.
And what codec will be used

```
 F4 ACK Alice -> Bob

   ACK sip:bob@client.biloxi.example.com SIP/2.0
   Via: SIP/2.0/TCP client.atlanta.example.com:5060;branch=z9hG4bK74bd5
   Max-Forwards: 70
   From: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
   To: Bob <sip:bob@biloxi.example.com>;tag=8321234356
   Call-ID: 3848276298220188511@atlanta.example.com
   CSeq: 1 ACK
   Content-Length: 0

   /* RTP streams are established between Alice and Bob */
```

SIP Basic Call Flow

```
/* Bob Hangs Up with Alice. Note that the CSeq is NOT 2, since
      Alice and Bob maintain their own independent CSeq counts.
      (The INVITE was request 1 generated by Alice, and the BYE is
      request 1 generated by Bob) */

 F5 BYE Bob -> Alice

   BYE sip:alice@client.atlanta.example.com SIP/2.0
   Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7
   Max-Forwards: 70
   From: Bob <sip:bob@biloxi.example.com>;tag=8321234356
   To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
   Call-ID: 3848276298220188511@atlanta.example.com
   CSeq: 1 BYE
   Content-Length: 0


 F6 200 OK Alice -> Bob

   SIP/2.0 200 OK
   Via: SIP/2.0/TCP client.biloxi.example.com:5060;branch=z9hG4bKnashds7;received=192.0.2.201
   From: Bob <sip:bob@biloxi.example.com>;tag=8321234356
   To: Alice <sip:alice@atlanta.example.com>;tag=9fxced76sl
   Call-ID: 3848276298220188511@atlanta.example.com
   CSeq: 1 BYE
   Content-Length: 0
```

SIP Basic Headers

Example INVITE from LCR to FreeSWITCH

```
INVITE sip:39525@127.0.0.1:5060 SIP/2.0.
Via: SIP/2.0/UDP 127.0.0.1:5050;rport;branch=z9hG4bK22j6Htm93t6yp.
Max-Forwards: 70.
From: <sip:41350@127.0.0.1:5050>;tag=04NX2vBmyjmDB.
To: <sip:39525@127.0.0.1:5060>.
Call-ID: 9522d949-c14e-1231-a5b0-00224da15b52.
CSeq: 898163649 INVITE.
Contact: <sip:127.0.0.1:5050>.
User-Agent: sofia-sip/1.12.11devel.
Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, NOTIFY, INFO.
Supported: timer, 100rel.
Content-Type: application/sdp.
Content-Length: 173.
.
v=0
o=LCR-Sofia-SIP 0 0 IN IP4 127.0.0.1
s=SIP Call
c=IN IP4 127.0.0.1
t=0 0
m=audio 34432 RTP/AVP 96 97 3
a=rtpmap:96 AMR/8000
a=rtpmap:97 GSM-EFR/8000
a=rtpmap:3 GSM/8000
```

SIP Basic Headers

- The first line of the SIP INVITE request is the Request-Line composed as:

  Method SP Request-URI SP SIP-Version CRLF
- In the example the method is INVITE, the request URI is sip:39525@127.0.0.1:5060 and SIP version is 2

We now have our set of headers

- **Via**: It contains the local address of 41350 where it's expecting the responses to come
- **Max-Forwards**: It's used to limit the number of hops that this request may take before reaching the recipient. It's decreased by one at each hop. It's necessary to prevent the request from traveling forever in case it is trapped in a loop.
- **From**: It also contains a display name of the user “41350” and a SIP or SIPS or TEL URI <sip:41350@127.0.0.1:5050>. It also contains a tag which is a pseudo-random sequence inserted by SIP application. It works s an identifier of the caller in the dialog

SIP Basic Headers

- **To**: It contains the display name and a SIP or SIPS URI
- **Call-ID**: It's a globally unique identifier of the call. The Call-ID is unique for a call. A call may contain several dialogs. Each dialog is uniquely identified by a combination of From, To and Call-ID
- **CSeq**: It contains an integer and a method name. When a transaction starts, the first message is given a random Cseq. After that it is incrementedby one with each new message. It used to detect non-delivery of a message or out-of-order messages delivery.
- **Contact**: It contains the URI that is a direct route to user 39525
- **User-Agent**: It's a string that identify what software as generated the request
- **Allow**: What type of SIP methods the client/server that has generated the request supports

SIP Basic Headers

- **Supported**: What capabilities the client/server supports
- **Content-Type**: It contains a description of the message body
- **Content-Length**: It's an octect (bytes) count of the message body

- Another important header is the **Record-Route** header, not shown in this example. The Record-Route header is inserted by SIP Proxy that wish to remain on the signalling path for the duration of the dialogue. For example for accounting/billing

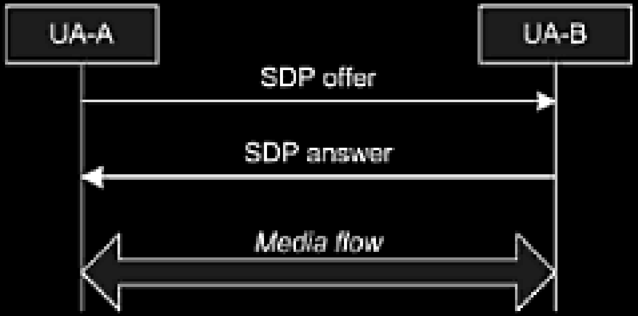
SDP Session Description Protocol

Offer in the INVITE

```
v=0
o=LCR-Sofia-SIP 0 0 IN IP4 127.0.0.1
s=SIP Call
c=IN IP4 127.0.0.1
t=0 0
m=audio 34432 RTP/AVP 96 97 3
a=rtpmap:96 AMR/8000
a=rtpmap:97 GSM-EFR/8000
a=rtpmap:3 GSM/8000
```

Answer in the 200 OK

```
v=0
o=LCR-Sofia-SIP 0 0 IN IP4 127.0.0.1
s=SIP Call
c=IN IP4 127.0.0.1
t=0 0
m=audio 34438 RTP/AVP 3
a=rtpmap:3 GSM/8000
```



- SDP Includes:
  - Session name and purpose
  - Time(s) the session is active
  - The media for the session and information to receive the media

SDP Session Description Protocol

Session description

- **v=** protocol version
- **o=** owner/creator session identifier
- **s=** session name
- **c=** connection information

Time description

- **t=** time the session is active

Media description

- **m=** media name and transport address
- **a=** zero or more media attributes

SIP Transactions/Dialogs

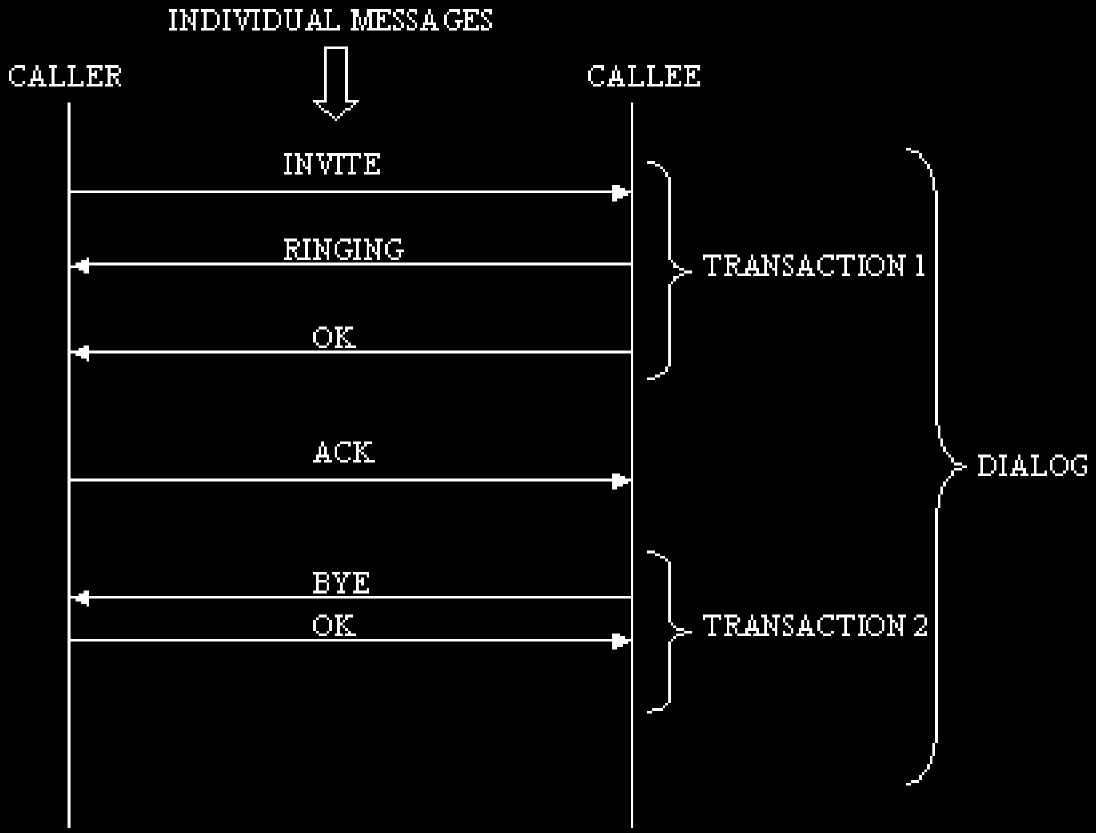
- In SIP we can talk about Calls, Dialogs,Transactions and Messages
- **Messages**: Are the individual textual bodies exchanged between a server and a client. There can be two type of Messages. Requests and Responses.

- **Transaction**: Occurs between client and server and involves all the messages from the first request sent from the client to the server up to a final (non 1xx) response sent from the client to the server. If the request is INVITE and final response is a non 2xx, the transaction also includes an ACK to the response.

- **Dialog**: Is a peer-to-peer SIP relationship between two Uas that persists for some time. A dialog is identified by a Call-ID, local tag and a remote tag.

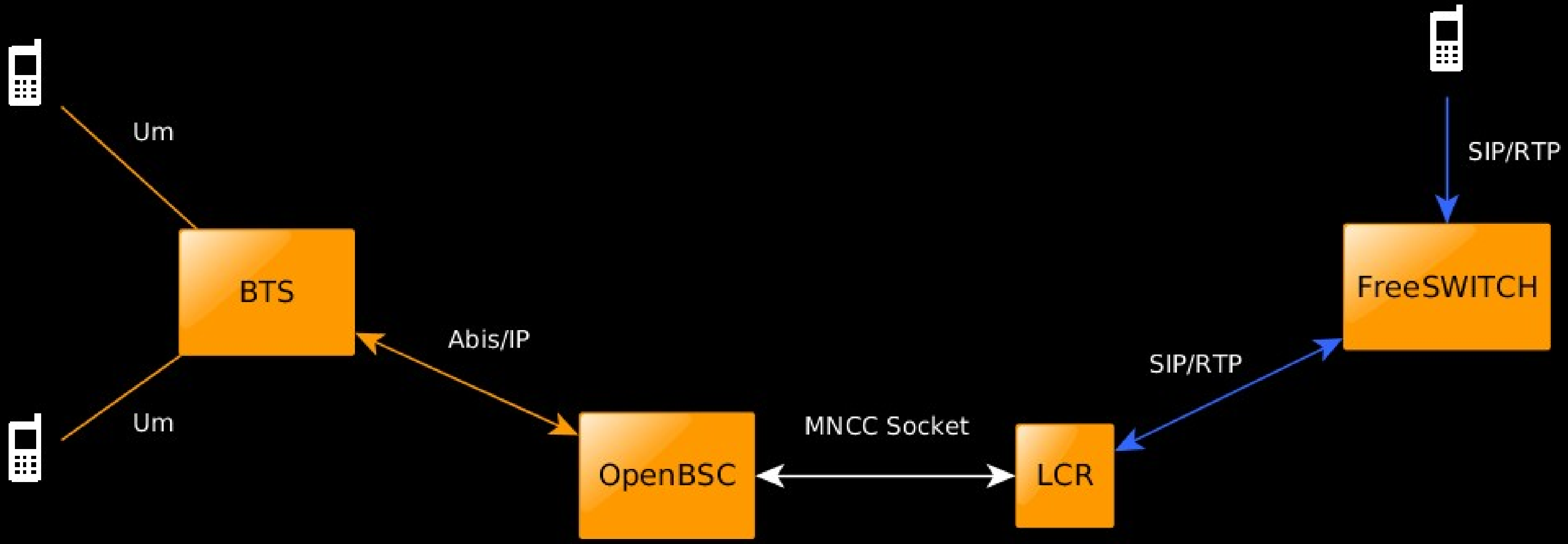
- **Call**: A call is composed by all the dialogs it's involved in.

SIP Transactions/Dialogs



RCN Simplified Diagram

Rhizomatica Community Network (RCN) Architecture


Um
SIP/RTP
FreeSWITCH
BTS
Abis/IP
SIP/RTP
Um
MNCC Socket
OpenBSC
LCR