

Seagull - SIP protocol

Table of contents

1 SIP protocol details.....	2
2 Getting started with SIP.....	2
2.1 First try.....	2
2.1.1 First try explained using external file.....	2
2.1.2 First try explained with configuration parameters.....	6
3 Miscellaneous considerations.....	9
4 SIP authentication.....	9
5 SIP Correlation example.....	10

1. SIP protocol details

The implementation of SIP in Seagull conforms to [SIP standards](http://www.packetizer.com/voip/sip/standards.html) (<http://www.packetizer.com/voip/sip/standards.html>)

Note:

Seagull doesn't currently support "optional" messages. The messages actually exchanged must exactly match the scenario.

2. Getting started with SIP

2.1. First try

So that you can get familiar with Seagull in the context of SIP, here is an example that will launch the following SIP scenario:

```
(1) INVITE
----->
(2) 180 Ringing
<-----
(3) 200 OK
<-----
(4) ACK
----->
(5) PAUSE
(6) BYE
----->
(7) 200 OK
<-----
```

2.1.1. First try explained using external file

Here is the commented version of the SIP client using a external file to set fields

Scenario. You can also see the client XML scenario.	Comments
t/viewvc/gull/seagull/trunk/src/exe-env/sip-env/scen	
<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <scenario> <counter> <counterdef name="callid-counter" init="0"> </counterdef> </counter> <traffic> <send channel="channel-1"> <action> <inc-counter name="callid-counter"> </inc-counter> <set-value name="call-id" format="\$(callid-counter)@255.255.255</pre>	<p>XML header</p> <p>The call id counter is declared and will be incremented for each new call</p> <p>For each new call, increment the call-id and set the call-id field</p>

<pre> <restore-from-external field="1" entity="call-id-addr"></restore-from- </pre>	<pre> Get address from the external file Store it for other fields </pre>
<pre> <store name="local_ip" entity="call-id-addr"></store> </pre>	<pre> Set the VIA field </pre>
<pre> <set-value name="via" format="SIP/2.0/UDP 255.255.255.255"></set-value> <restore name="local_ip" entity="via-addr"></restore> </pre>	<pre> Get protocol from external file Set the From field </pre>
<pre> <restore-from-external field="4" entity="via-transport"></restore-from- </pre>	<pre> Get Name from external file Store it for other fields </pre>
<pre> <set-value name="from" format="seagull <sip:seagull@255.255.255.255&gt;;tag= </pre>	<pre> Set the To field </pre>
<pre> <restore-from-external field="0" entity="from-name"></restore-from-ext </pre>	<pre> Get Name from external file </pre>
<pre> <store name="local_name" entity="from-name"></store> </pre>	<pre> Get address from external file </pre>
<pre> <restore name="local_ip" entity="from-addr"></restore> </pre>	<pre> Set the INVITE request address and name </pre>
<pre> <set-value name="to" format="sut <sip:service@255.255.255.255:5060&gt;; </pre>	<pre> Header </pre>
<pre> <restore-from-external field="2" entity="to-name"></restore-from-exte </pre>	<pre> Body </pre>
<pre> <restore-from-external field="3" entity="to-addr"></restore-from-exte </pre>	
<pre> <restore name="local_name" entity="invite-name"></restore> <restore name="local_ip" entity="invite-addr"></restore> </pre>	
<pre> </action> <message> </pre>	
<pre> <! [CDATA[INVITE sip:schooler@cs.caltech.edu SIP/2.0 Via: SIP/2.0/UDP north.east.isi.edu From: Mark Handley <mailto:sip:mjh@isi.edu> To: Eve Schooler </pre>	<pre> Store completed fields for future msg </pre>

<pre><sip:schooler@caltech.edu> Call-ID: 2963313058@north.east.isi.edu CSeq: 1 INVITE Subject: SIP will be discussed, too Content-Type: application/sdp Content-Length: 187]] ></pre>	Header
<pre><![CDATA[v=0 o=user1 53655765 2353687637 IN IP4 155.208.221.173 s=Mbone Audio i=Discussion of Mbone Engineering Issues e=mbone@somewhere.com c=IN IP4 155.208.221.173 t=0 0 m=audio 3456 RTP/AVP 0 a=rtptime:0 PCMU/8000]] ></pre>	Header
<pre></message> <action> <store name="SID" entity="call-id"></store> <store name="VIA" entity="via"></store> <store name="FROM" entity="from"></store> <store name="TO" entity="to"></store> </action> </send></pre>	Restore fields
<pre><receive channel="channel-1"> <message></pre>	Set name and address in the Ack request
<pre><![CDATA[SIP/2.0 180 Ringing Via: SIP/2.0/UDP north.east.isi.edu From: Mark Handley <sip:mjh@isi.edu> To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472 Call-ID: 2963313058@north.east.isi.edu CSeq: 1 INVITE]] > </message> </receive></pre>	Header
<pre><receive channel="channel-1"> <message></pre>	Restore fields
<pre><![CDATA[SIP/2.0 200 OK Via: SIP/2.0/UDP north.east.isi.edu From: Mark Handley <sip:mjh@isi.edu> To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472 Call-ID:</pre>	Set name and address in the BYE request
<pre></pre>	Header

```
2963313058@north.east.isi.edu
      CSeq: 1 INVITE
      Contact:
sip:es@jove.cs.caltech.edu]] >
  </message>
</receive>
<send channel="channel-1">
  <action>

    <restore name="SID"
entity="call-id"></restore>
    <restore name="VIA"
entity="via"></restore>
    <restore name="FROM"
entity="from"></restore>
    <restore name="TO"
entity="to"></restore>

    <restore name="local_name"
entity="ack-name"></restore>
    <restore name="local_ip"
entity="ack-addr"></restore>
  </action>
  <message>

    <! [ CDATA[ACK
sip:es@jove.cs.caltech.edu SIP/2.0
      Via: SIP/2.0/UDP
north.east.isi.edu
      From: Mark Handley
<sip:mjh@isi.edu>
      To: Eve Schooler
<sip:schooler@caltech.edu>
;tag=9883472
      Call-ID:
2963313058@north.east.isi.edu
      CSeq: 1 ACK]] >
  </message>
</send>

  <send channel="channel-1">
  <action>

    <restore name="SID"
entity="call-id"></restore>
    <restore name="VIA"
entity="via"></restore>
    <restore name="FROM"
entity="from"></restore>
    <restore name="TO"
entity="to"></restore>

    <restore name="local_name"
entity="bye-name"></restore>
    <restore name="local_ip"
entity="bye-addr"></restore>
  </action>
  <message>

    <! [ CDATA[BYE
sip:watson@y.bell-tel.com SIP/2.0
      Via: SIP/2.0/UDP
c.bell-tel.com
      From: A. Bell
<sip:a.g.bell@bell-tel.com>
      To: T. Watson
```

Header

```

<sip:t.watson@ieee.org>
;tag=35253448
    Call-ID:
2963313058@north.east.isi.edu
    CSeq:    2 BYE]] >
</message>
</send>

<receive channel="channel-1">
<message>

    <! [CDATA[SIP/2.0 200 OK
    Via:    SIP/2.0/UDP
c.bell-tel.com
    From:    A. Bell
<sip:a.g.bell@bell-tel.com>
    To:    T. Watson
<sip:t.watson@ieee.org>
;tag=35253448
    Call-ID:
2963313058@north.east.isi.edu
    CSeq:    2 BYE]] >
</message>
</receive>

</traffic>

</scenario>

```

2.1.2. First try explained with configuration parameters

Here is the commented version of the SIP client using configuration parameters to set fields

In the configuration file, several parameters are added, for example:

```

<define entity="config-param" name="param_call-id-addr"
value="127.0.0.1:5060"></define>
<define entity="config-param" name="param_via" value="SIP/2.0/UDP
127.0.0.1"></define>
<define entity="config-param" name="param_from" value="seagull
<sip:seagull@127.0.0.1:5060>;tag=1"></define>
<define entity="config-param" name="param_to" value="sut
<sip:service@127.0.0.1:5060"></define>

```

They will be used in the scenario.

Scenario. You can also see the client XML scenario .	Comments
<pre> <?xml version="1.0" encoding="ISO-8859-1" ?> <scenario> <counter> <counterdef name="callid-counter" init="0"> </counterdef> </counter> <traffic> <send channel="channel-1"> <action> <!-- For each new call, increment the callid counter --> <inc-counter </pre>	<pre> XML header The call id counter is declared and will be incremented for each new call First scenario command: send a INVITE message Action done before sending message For each new call, increment the call-id and set the call-id field </pre>

<pre> name="callid-counter"> </inc-counter> <!-- And set the call-id field --> <set-value name="call-id" format="\$(callid-counter)@127.0.0.1"> </action> <message> <!-- header --> <! [CDATA[INVITE sip:\$(param_call-id-addr) SIP/2.0 Via: \$(param_via) From: \$(param_from) To: \$(param_to) Call-ID: 2963313058@north.east.isi.edu CSeq: 1 INVITE Subject: SIP will be discussed, too Content-Type: application/sdp Content-Length: 187]] > <!-- body --> <![CDATA[v=0 o=user1 53655765 2353687637 IN IP4 155.208.221.173 s=Mbone Audio i=Discussion of Mbone Engineering Issues e=mbone@somewhere.com c=IN IP4 155.208.221.173 t=0 0 m=audio 3456 RTP/AVP 0 a=rtptime:0 PCMU/8000]] > </message> <action> <!-- Store completed fields for future msg--> <store name="SID" entity="call-id"></store> </action> </send> <receive channel="channel-1"> <message> <!-- header --> <! [CDATA[SIP/2.0 180 Ringing Via: SIP/2.0/UDP north.east.isi.edu From: Mark Handley <sip:mjh@isi.edu> To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472 Call-ID: 2963313058@north.east.isi.edu CSeq: 1 INVITE]] > </message> </receive> <receive channel="channel-1"> <message> <!-- header --> </pre>	<pre> Set the INVITE request address and name Header Set address from the configuration parameter "param_call-id-addr" Set Via from the configuration parameter "param_via" Set From from the configuration parameter "param_from" Set To from the configuration parameter "param_to" Body Action done after sending message Store session-id ("Call-id") field for future msg scenario command: receive a 180 message Header scenario command: receive a 200 message Header </pre>
---	--

<pre> <! [CDATA[SIP/2.0 200 OK Via: SIP/2.0/UDP north.east.isi.edu From: Mark Handley <sip:mjh@isi.edu> To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472 Call-ID: 2963313058@north.east.isi.edu CSeq: 1 INVITE Contact: sip:es@jove.cs.caltech.edu]] > </message> </receive> </pre>	<pre> scenario command: send a ACK message Restore session-id in the "Call-id" field Header </pre>
<pre> <send channel="channel-1"> <action> <!-- restore fields --> <restore name="SID" entity="call-id"></restore> </action> <message> <!-- header --> <! [CDATA[ACK sip:\$(param_call-id-addr) SIP/2.0 Via: \$(param_via) From: \$(param_from) To: \$(param_to) Call-ID: 2963313058@north.east.isi.edu CSeq: 1 ACK]] > </message> </send> </pre>	<pre> scenario command: send a BYE message Restore session-id in the "Call-id" field Header </pre>
<pre> <send channel="channel-1"> <action> <!-- restore fields --> <restore name="SID" entity="call-id"></restore> </action> <message> <!-- header --> <! [CDATA[BYE sip:\$(param_call-id-addr) SIP/2.0 Via: \$(param_via) From: \$(param_from) To: \$(param_to) Call-ID: 2963313058@north.east.isi.edu CSeq: 2 BYE]] > </message> </send> </pre>	<pre> scenario command: receive a 200 message Header </pre>
<pre> <receive channel="channel-1"> <message> <!-- header --> <! [CDATA[SIP/2.0 200 OK Via: SIP/2.0/UDP c.bell-tel.com From: A. Bell <sip:a.g.bell@bell-tel.com> To: T. Watson <sip:t.watson@ieee.org>;tag=35253448 Call-ID: 2963313058@north.east.isi.edu CSeq: 2 BYE]] > </pre>	


```

</message>
</receive>

</traffic>

</scenario>

```

3. Miscellaneous considerations

Users must be aware of the fact that Seagull currently works only with "perfect" SIP scenarios. It means that the following messages are not properly handled:

- Retransmissions (in reception)
- INFO and NOTIFY messages
- Optional responses

These points are currently in progress.

4. SIP authentication

Seagull supports two authentication algorithm: Digest/MD5 ("algorithm="MD5") and Digest/AKA ("algorithm="AKAv1-MD5").

Computing the authorization header is done through the usage of the "method" in a "set-value" action in the scenario.

Depending on the algorithm ("MD5" or "AKAv1-MD5"), different parameters must be passed next to the www-authentication field:

- Digest/MD5:
 - **username:** username
 - **password:** password
 - **auth:** authentication parameters (example auth=Digest realm="cosims.net", nonce="b8Q2u9ZxyE3VKg6ZD3P07Wo6myth4AAArn7NvQIWWR4=", algorithm=MD5)
 - **method:** SIP method (example method=INVITE)
 - **uri:** Uniform Resource Identifier (example uri=sip:127.0.0.1:5061)
- Digest/AKA:
 - **username:** username (example : username=joe)
 - **aka_k:** Permanent secret key (example : aka_k=0x465B5CE8B199B49FAA5F0A2EE238A6BC)
 - **aka_op:** OPerator variant key (example : aka_op=0xCDC202D5123E20F62B6D676AC72CB318)
 - **aka_amf:** Authentication Management Field (indicates the algorithm and key in use) (example aka_amf=0xB9B9)
 - **auth:** authentication parameters (example auth=Digest realm="cosims.net", nonce="b8Q2u9ZxyE3VKg6ZD3P07Wo6myth4AAArn7NvQIWWR4=", algorithm=AKAv1-MD5)
 - **method:** SIP method (example method=INVITE)
 - **uri:** Uniform Resource Identifier (example uri=sip:127.0.0.1:5061)

In case of authentication (example for AKA):

Example of scenario which get external data from a file:

```

<send channel="channel-1">
  <action>
    <!-- For each new call, increment the callid counter -->
    <inc-counter name="callid-counter"> </inc-counter>

```

```

<set-value name="call-id"
           format="$(callid-counter)@255.255.255.255"></set-value>

<!-- Restore data for authentication -->
<restore-from-external field="0" name="USER-AUTH"></restore-from-external>
<restore-from-external field="1" name="AUTH-AUTH"></restore-from-external>
<restore-from-external field="2" name="METHOD-AUTH"></restore-from-external>
<restore-from-external field="3" name="URI-AUTH"></restore-from-external>
<restore-from-external field="4" name="OP-AUTH"></restore-from-external>
<restore-from-external field="5" name="K-AUTH"></restore-from-external>
<restore-from-external field="6" name="AMF-AUTH"></restore-from-external>
<set-value name="www-authentication" method="authentication"
format="username=$(USER-AUTH);aka_op=$(OP-AUTH);aka_k=$(K-AUTH);aka_amf=$(AMF-AUTH);auth=$(AUTH-AUTH)"></set-value>

</action>
</message>
<!-- header -->
<![CDATA[INVITE sip:schooler@cs.caltech.edu SIP/2.0
Via: SIP/2.0/UDP north.east.isi.edu
From: Mark Handley <sip:mjh@isi.edu>
To: Eve Schooler <sip:schooler@caltech.edu>
Call-ID: 2963313058@north.east.isi.edu
CSeq: 1 INVITE
Subject: SIP will be discussed, too
Content-Type: application/sdp
Content-Length: 187]] >
<!-- body -->
<![CDATA[v=0
o=user1 53655765 2353687637 IN IP4 155.208.221.173
s=Mbone Audio
i=Discussion of Mbone Engineering Issues
e=mbone@somewhere.com
c=IN IP4 155.208.221.173
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000]] >
</message>
<action>
<!-- Store completed fields for future msg-->
<store name="SID" entity="call-id"></store>
<store name="VIA" entity="via"></store>
<store name="FROM" entity="from"></store>
<store name="TO" entity="to"></store>
</action>
</send>

```

The external data file looks like that:

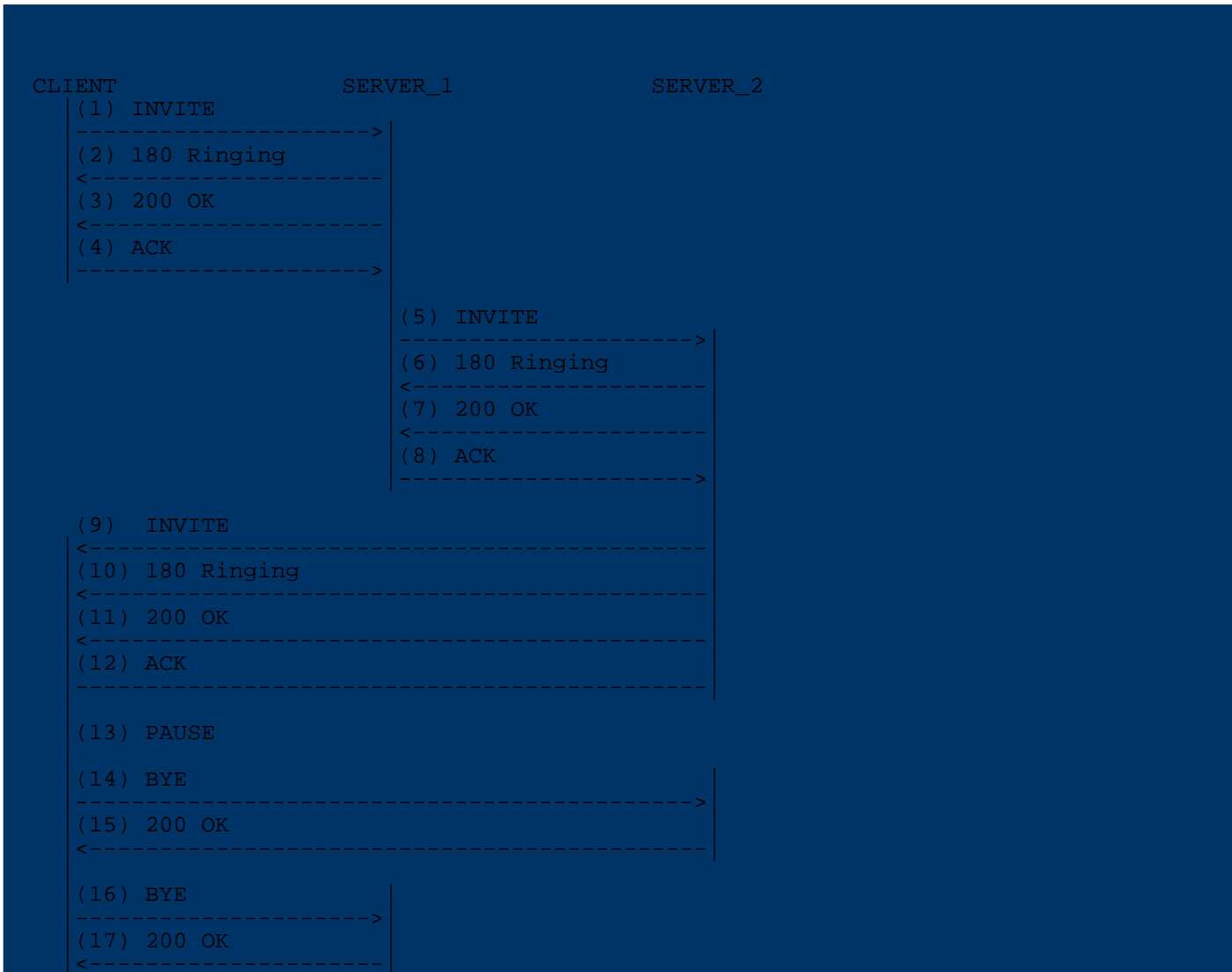
```

"string" ; "string"
; "string" ; "string" ; "string" ; "string"
; "string" ;
"joe" ; "Digest realm=\"cosims.net\",
nonce=\"b8d2v9ZxfE3VHg6ZD3P87Wolmfth4BBArn7VFQlWWR4=\", algorithm=AKAv1-MD5" ; "INVITE"
; "sip:127.0.0.1:5061" ; "0xCDC2458FEB3E20F62B6D676AC72CB318" ;
"0x465B5CE8B199B4BA48D2B72EE238A6BC" ; "0xB9A3" ;
"xxxxxx" ; "Digest xxxxxxxx"
" ; "XXXXXX" ; "xxxxxxxxxxxxxxxxxxxxxx" ; "0xAAAAAAAAAAAAAAAAAAAAAAAAAAAA" ;
"0xAAAAAAAAAAAAAAAAAAAAAAAAAAAA" ; "0xAAAA" ;
...
...

```

5. SIP Correlation example

An example of the correlation feature is proposed in the SIP environment. Here is the call flow:



The call-id used between the client and the server_1 is defined by the client. The call-id used between the server_1 and the server_2 and between the server_2 and the client is defined by the server_1. The value of this last call-id is sent from the server_1 to the client in the "o" field of the body of the "200 OK" (3) message.

To execute this example, Open three terminal sessions. Terminal 2 will be the server_1, Terminal 3 will be the server_2 and Terminal 1 the client. Examples are located in the "run" directory. So the first thing you need to do is to go in this directory (in both terminal windows):

```
cd run
```

In Terminal 2 window type:

```
./start_server_corr_1.ksh
```

In Terminal 3 window type:

```
./start_server_corr_2.ksh
```

In Terminal 1 window type:

```
./start_client_corr.ksh
```