

Seagull - H248/MEGACO protocol

Table of contents

1 H248/MEGACO protocol details.....	2
2 Getting started with H248.....	2
2.1 First try.....	2
2.2 First try explained.....	2
3 Miscellaneous considerations.....	7

1. H248/MEGACO protocol details

The implementation of H248/MEGACO in Seagull conforms to [H248 standards](http://www.packetizer.com/voip/h248/standards.html) (<http://www.packetizer.com/voip/h248/standards.html>), ASCII form (ASN.1 version not supported).

Note:

Due to the nature of H248 (one scenario contains several H248 transactions with multiple transaction-ids), Seagull supports only **client** scenarios. Seagull supports server scenarios with the limitation of one H248 transaction per scenario.

Note:

H248 implementation in Seagull can be entirely customized through the H248 XML dictionary - which should allow almost any H248 extensions to be supported.

2. Getting started with H248

2.1. First try

Note:

As a generic rule, Seagull is packaged with default server and client scenario so that you can start to derive test scenarios from the one already embedded. In the case of H.248, the default client (client.xml) and server (server.xml) scenario are not H.248 compliant in the sense that the transaction-id is the same for all transactions during a scenario. As Seagull supports H248 "client" scenarios with multiple transaction-ids, and to provide an H248 compliant example, another scenario called "client_multi_tid.xml" is provided but cannot be executed against a Seagull server. The following section details this "client_multi_tid.xml" scenario.

So that you can get familiar with Seagull in the context of H248, here is an example that will launch the following H248 scenario:

```

Trans (Add)
----->
Reply (Add)
<-----
TransRespAck
----->
Trans (Play)
----->
Trans (Sub)
----->
Reply (Sub)
<-----
TransRespAck
----->
    
```

You can also see the [client_multi_tid](#) (h248.client_multi_tid.xml.html) XML scenario.

2.2. First try explained

Here is the commented version of the H248 client which, for the example, mixes both short and pretty H248 form:

Scenario	Comments
<pre> <?xml version="1.0" encoding="ISO-8859-1" ?> <scenario> <counter> </pre>	<pre> XML header The transaction-counter counter is declared and </pre>

```

<counterdef
name="transaction-counter"
init="999">
  </counterdef>
</counter>

<traffic>

<!--##### ADD
##----->
<!-------Send
ADD----->
  <send channel="channel-1">
    <action>
      <inc-counter
name="transaction-counter"></inc-counter>
      <set-value
name="transaction-id"
format="$(transaction-counter)"></set-value>
    </action>
    <message>
      <!-- header -->
      <![CDATA[!/1
[16.16.88.188\]:55554
          T=18571]] >
      <!-- body -->
      <![CDATA[C=${A=${M{TS{SI=iv,BF=off},
ST=1{O{MO=sr,RV=off,RG=off},
          R{m=audio 49152
RTP/AVP 3 97 98 8 0 101
          c=IN IP4
16.16.214.175
          a=rtpmap:3
GSM/8000
          a=rtpmap:97
iLBC/8000
          a=rtpmap:98
iLBC/8000
          a=fmtp:98 mode=20
          a=rtpmap:8
PCMA/8000
          a=rtpmap:0
PCMU/8000
          a=rtpmap:101
telephone-event/8000
          a=fmtp:101 0-11,16
          }]}]}]}]] >
    </message>
    <action>
      <!-- Store the
transaction-id for following
messages -->
      <store name="TID"
entity="transaction-id">
        <regexp name="t-id"
expr="[Tt][[:space:]]*=[[:space:]]*"
          nbexpr="2"
          subexpr="1"></regexp>
      </store>
    </action>
  </send>
<!-------Receive ADD
reply----->
  <receive channel="channel-1">
    <message>
      <!-- header -->

```

will be incremented for each new transaction (several time per call)

Send the ADD

Increment the transaction counter ...and set the transaction-id field, as declared in the dictionary with the transaction-counter value

H248 header. Note the "\]" instead of "]"

The value of the Transaction-Id (T=18571 here) is replaced by Seagull by the previous action.

H248 body. Note that the first "{" is omitted (see note below)

Now that the message is sent, we store the value of the transaction ID (the counter is global to the entire traffic and thus its value needs to be saved)

Receive the Reply on channel-1

Note that the content of the message is not checked by Seagull. Content check can be done using the check-value action (see core documentation)

```

<![CDATA[MEGACO/1
[16.16.88.188\]:55555
      Reply=18571]] >
  <!-- body -->
<![CDATA[Context=1{Add=T1{Media
      {TerminationState{
ServiceStates=inservice,Buffer=off},
Stream=1{LocalControl{Mode=sendreceive,
dValue=off,ReservedGroup=off},Local{
      v=0
      o=user 0 0 IN IP4
16.16.88.188
      s=-
      t=0 0
      m=audio 3436
RTP/AVP 8 0 101
      c=IN IP4
16.16.88.188
      a=rtpmap:8
PCMA/8000/1
      a=rtpmap:0
PCMU/8000/1
      a=rtpmap:101
telephone-event/8000/1
      },
      Remote{
      v=0
      o=user 0 0 IN IP4
16.16.88.188
      s=-
      t=0 0
      m=audio 49152
RTP/AVP 3 97 98 8 0 101
      c=IN IP4
16.16.214.175
      a=rtpmap:3
GSM/8000
      a=rtpmap:97
iLBC/8000
      a=rtpmap:98
iLBC/8000
      a=fmtp:98 mode=20
      a=rtpmap:8
PCMA/8000
      a=rtpmap:0
PCMU/8000
      a=rtpmap:101
telephone-event/8000
      a=fmtp:101 0-11,16
      }]]]]]]]] >
</message>
<action>
  <!-- Store the
termination-id for following
messages -->
  <store name="TERMID"
entity="termination-id"></store>
  <!-- Store the context-id
for following messages -->
  <store name="CONTID"
entity="context-id">
  <regexp name="cont-id"
expr="Context[[:space:]]*=[ ]([[:digit:]]
nbexpr="2"
subexpr="1"></regexp>
  </store>

```

Termination-Id and Context-Id are stored from the message received so that they can be re-used later in the scenario

Send the TransactionResponseAck. For this message, the transaction id is located in the body. This is why you must use the 'restore' action on the 'transaction-id-in-body' field. Note that fields are described in the dictionary and correspond to regular expressions.

Send the Modify

```

    </action>
  </receive>
<!-------Send
Ack----->
  <send channel="channel-1">
    <action>
      <restore name="TID"
entity="transaction-id-in-body">
      </restore>
    </action>
    <message>
      <!-- header -->
      <![CDATA[!/1
[16.16.88.188\]:55554
      TransactionResponseAck]]
>
      <!-- body -->
      <![CDATA[18571]] >
    </message>
  </send>

<!-------## MODIFY
##----->
<!-------Send
MODIFY----->
  <send channel="channel-1">
    <action>
      <inc-counter
name="transaction-counter"></inc-cour
      <set-value
name="transaction-id"
format="$(transaction-counter)"></set
      <!-- Seagull's session ID
was value of TID and
      is now value of
transaction-id -->
      <set-new-session-id
name="TID" entity="transaction-id">
      </set-new-session-id>
      <restore name="TERMID"
entity="termination-id"></restore>
      <restore name="CONTID"
entity="context-id"></restore>
    </action>
    <message>
      <!-- header -->
      <![CDATA[!/1
[16.16.88.188\]:55554
      Transaction=18572]]
>
      <!-- body -->
      <![CDATA[C=1{MF=T1
      {SG{aasb/play
{an="sid=<file://home/ocmpadm/welcome
      },
      E=100{g/sc}}}}]]
>
    </message>
    <action>
      <!-- Store the
transaction-id for following
messages -->
      <store name="TID"
entity="transaction-id">
      <regexp name="t-id"
expr="Transaction*[-][[:space:]]*([0-
```

Increment the transaction-counter
 Set the value of transaction-id
 field with the new value
 of the transaction counter. Make
 Seagull aware that the
 scenario execution that was
 identified with transaction-id "TID"
 is now identified with the value of
 transaction-id field.
 Re-use the termination-id and
 context-id stored earlier

The value of Transaction is replaced
 by the set-value action

The value of context and
 termination-id are replaced by the
 restore actions

Just after sending the message with
 the new transaction-id, store
 this transaction-id for later use in
 the scenario

```

        nbexpr="2"
        subexpr="1"></regexp>
    </store>
</action>
</send>
<!-------Receive MODIFY
response----->
    <receive channel="channel-1">
        <message>
            <!-- header -->
            <![CDATA[MEGACO/1
[16.16.88.188\]:55555
                Reply=18572]] >
            <!-- body -->
<![CDATA[Context=1{Modify=T1}]] >
        </message>
    </receive>
    <!-------Send MODIFY
Ack----->
    <send channel="channel-1">
        <action>
            <restore name="TID"
entity="transaction-id-in-body">
            </restore>
        </action>
        <message>
            <!-- header -->
            <![CDATA[!/1
[16.16.88.188\]:55554
TransactionResponseAck]] >
            <!-- body -->
            <![CDATA[18572]] >
        </message>
    </send>

    <!-------## SUBTRACT
##----->
    <!-------Send
SUBTRACT----->
    <send channel="channel-1">
        <action>
            <inc-counter
name="transaction-counter"></inc-cour
            <set-value
name="transaction-id"
format="$(transaction-counter)"></set
            <!-- Seagull's session ID
was value of TID and
            is now value of
transaction-id -->
            <set-new-session-id
name="TID" entity="transaction-id">
            </set-new-session-id>
            <restore name="CONTID"
entity="context-id"></restore>
            <restore name="TERMID"
entity="termination-id"></restore>
        </action>
        <message>
            <!-- header -->
            <![CDATA[!/1
[16.16.88.188\]:55554
                Transaction=18573]]
>
            <!-- body -->
            <![CDATA[C=1{S=T1}]] >

```

Increment the transaction-counter,
as done previously

```

    </message>
    <action>
    <!-- Store the transaction-id
for following messages -->
    <store name="TID"
entity="transaction-id">
    <regex name="t-id"
expr="Transaction*[[[:space:]]*([0-
    nbexpr="2"
    subexpr="1"></regex>
    </store>
    </action>
</send>
<!-------Receive SUBTRACT
reply----->
    <receive channel="channel-1">
    <message>
    <!-- header -->
    <![CDATA[MEGACO/1
[16.16.88.188\]:5555
    Reply=18573]] >
    <!-- body -->
<![CDATA[Context=1{Subtract=T1}]] >
    </message>
    </receive>
<!-------Send
Ack----->
    <send channel="channel-1">
    <action>
    <restore name="TID"
entity="transaction-id-in-body">
    </restore>
    </action>
    <message>
    <!-- header -->
    <![CDATA[!/1
[16.16.88.188\]:5554
TransactionResponseAck]] >
    <!-- body -->
    <![CDATA[18573]] >
    </message>
    </send>

</traffic>
</scenario>

```

3. Miscellaneous considerations

When creating the H248 scenario, several points must be considered:

- The "set-new-session-id" action allows to change how Seagull identifies the execution of the current scenario. In the H248 case, it used before each new transaction, when the transaction-id needs to change. See [Core documentation](#) (core.html#set-new-session-id) for explanations.
- In a CDATA section, "]" must be escaped by using "\]" - this does not apply to "[".
- The number of "{" and "}" in H248 messages is not equal: because of the way the dictionary is done, there must be one less "{" than the number of "}" (no starting "{" in the body, but ending "}" must be present).
- In H248 scenarios, heading spaces and tabs are "eaten" by the scenario parser and are not sent on the network.