Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2012

C. Jennings
S. Nandakumar
Cisco
C. Holmberg
Ericsson

October 30, 2011

SIP to RTCWeb Offer/Answer Protocol (ROAP) Gateway
draft-jennings-rtcweb-signaling-gateway-01

Abstract

This document proposes behavior of a RTCWeb signaling gateway for mapping message representations between RTCWeb Offer/Answer Protocol (ROAP) scheme and native SIP messaging scheme. Such a signaling gateway is intended to translate to and from/SIP for enabling use cases between a RTCWeb enabled browser and legacy SIP devices.

Table of Contents

---

1. Introduction

This specification suggests one possible way to build a RTCWeb Signaling gateway that maps message representations proposed in **[ROAP]** to native SIP **[RFC3261]** messages and vice-versa. The specification **[ROAP]** describes a signaling protocol for RTCWeb to support negotiation of media session using SDP offer/answer **[RFC3264]** protocol. Such a signaling protocol enables an RTCWeb browser to setup media sessions to another browser or a SIP device. For Browser-to-SIP device use case, the signaling gateway connects to legacy SIP devices and SHALL translate messages between ROAP and SIP native messages schemes.

2. Mapping to SIP

TOC

Note:
The SDP and SIP examples are not correct but illustrate the rough outline of the mechanism. Future version will correct this.

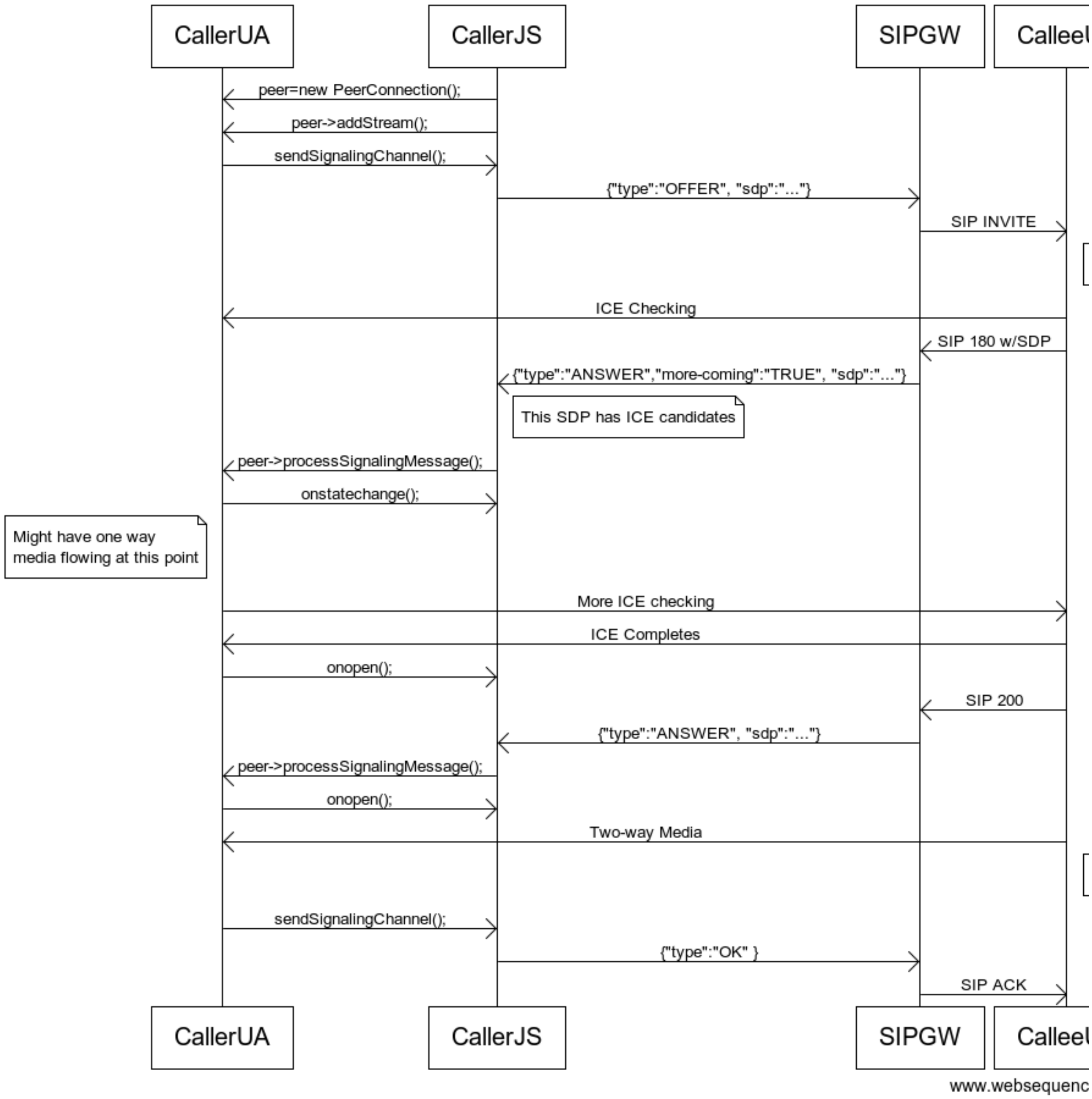
The design requires the gateway to be SIP transaction statefull but does not require any storage of longer term state. The information that remains constant over the SIP dialog is stored in session tokens while the information that is needed to from a SIP response is stored in response tokens. The gateway appears as a SIP UA to the sip side. Message on the two sides of the signalling gateway are referred to as the SIP side and web side.

The following sub-sections show example message flows with detailed message description of native SIP messages that are mapped from ROAP scheme and the ones that are received as responses by the signaling gateway. CallerUA(callerua@atlanta.example.com) is a RTCWeb browser. CalleeUA(sip:calleeua@sippy.example.com) is assumed to be a SIP-enabled device. It is also assumed that CalleeUA has registered with a SIP proxy server to be able to receive the calls via the proxy.

2.1. SuccessFull Session Establishment

TOC

In this scenario CallerUA establishes successful media session with CalleeUA, a legacy SIP end-point, with the help of the RTCWeb signaling gateway.



Message Details

Signaling gateway (on behalf of CallerUA) maps ROAP:OFFER (section 5.3.1 of ROAP**[ROAP]**) to SIP:INVITE and sends it to CalleeUA to start the session.

```
{"type":"OFFER",
 "offererSessionId":"36707f69b",
 "seq": 1
 "sdp":"
      v=0
      o=callerua 1429 0  IN IP4 client.atlanta.example.com
      s=Call
      c=IN IP4 192.0.2.101
      t=0 0
      m=audio 16384 RTP/AVP 0
      a=rtpmap:0 PCMU/8000
      a=sendrecv"
}
```

```
{INVITE sip:calleeua@sippy.example.com SIP/2.0
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: CallerUA <sip:callerua@atlanta.example.com>;tag=36707f69b
To: CalleeUA <sip:calleeua@sippy.example.com>
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:callerua@client.atlanta.example.com;transport=udp>
Content-Type: application/sdp

v=0
o=callerua 1429 0  IN IP4 client.atlanta.example.com
s=Call
c=IN IP4 192.0.2.101
t=0 0
m=audio 16384 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendrecv
}
```

SIP:180 from CalleeUA is received at the signaling gateway.

```
{SIP/2.0 180 Ringing
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=192.0.2.101
From: CallerUA <sip:callerua@atlanta.example.com>;tag=36707f69b
To: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
Call-ID:00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:calleeua@client.sippy.example.com;transport=udp>

v=0
o=calleeua 2890844527 2890844527 IN IP4 client.sippy.example.com
s=Call
c=IN IP4 192.0.2.201
t=0 0
m=audio 16834 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendrecv
}
```

This message SHALL be converted to ROAP:Answer (section 5.3.2 of ROAP**[ROAP]**) with "more-coming" flag set to true as described in the section 5.2.3 of ROAP**[ROAP]** specification and sent to CallerUA by the signaling gateway.

```
{"type":"ANSWER",
 "offererSessionId":"36707f69b",
 "answererSessionId":"8321234356",
 "seq": 1,
 "more-coming": true,
 "sdp":"
      v=0
      o=callerua 1429 0  IN IP4 client.atlanta.example.com
      s=Call
      c=IN IP4 192.0.2.101
      t=0 0
      m=audio 16384 RTP/AVP 0
      a=rtpmap:0 PCMU/8000
      a=sendrecv"
}
```

SIP: OK from CalleeUA is received at the signaling gateway.

```
{SIP/2.0 200 OK
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=192.0.2.101
From: CallerUA <sip:callerua@atlanta.example.com>;tag=36707f69b
To: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 1 INVITE
Contact: <sip:calleeua@client.sippy.example.com;transport=udp>
Content-Type: application/sdp

v=0
```

```
o=calleeua 2890844527 2890844527 IN IP4 client.sippy.example.com
s=Call
c=IN IP4 192.0.2.201
t=0 0
m=audio 16834 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendrecv
}
```

This message SHALL be converted to ROAP:Answer(section 5.3.2 of ROAP**[ROAP]**) and sent to caller by the signaling gateway. This represents a final answer as described in the section 5.2.3 of ROAP**[ROAP]**

```
 {"type":"ANSWER",
  "offererSessionId":"36707f69b",
  "answererSessionId":"8321234356",
  "seq": 1,
  "sdp":"
v=0
o=calleeua 2890844527 2890844527 IN IP4 client.sippy.example.com
s=Call
c=IN IP4 192.0.2.201
t=0 0
m=audio 16834 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendrecv"
 }
```

Signaling gateway (on behalf of CallerUA) maps ROAP:OK (section 5.3.2 of ROAP**[ROAP]**) to SIP:ACK and sends it to CalleeUA to start the session. This completes call-setup and media streams are established between CallerUA and the CalleeUA.

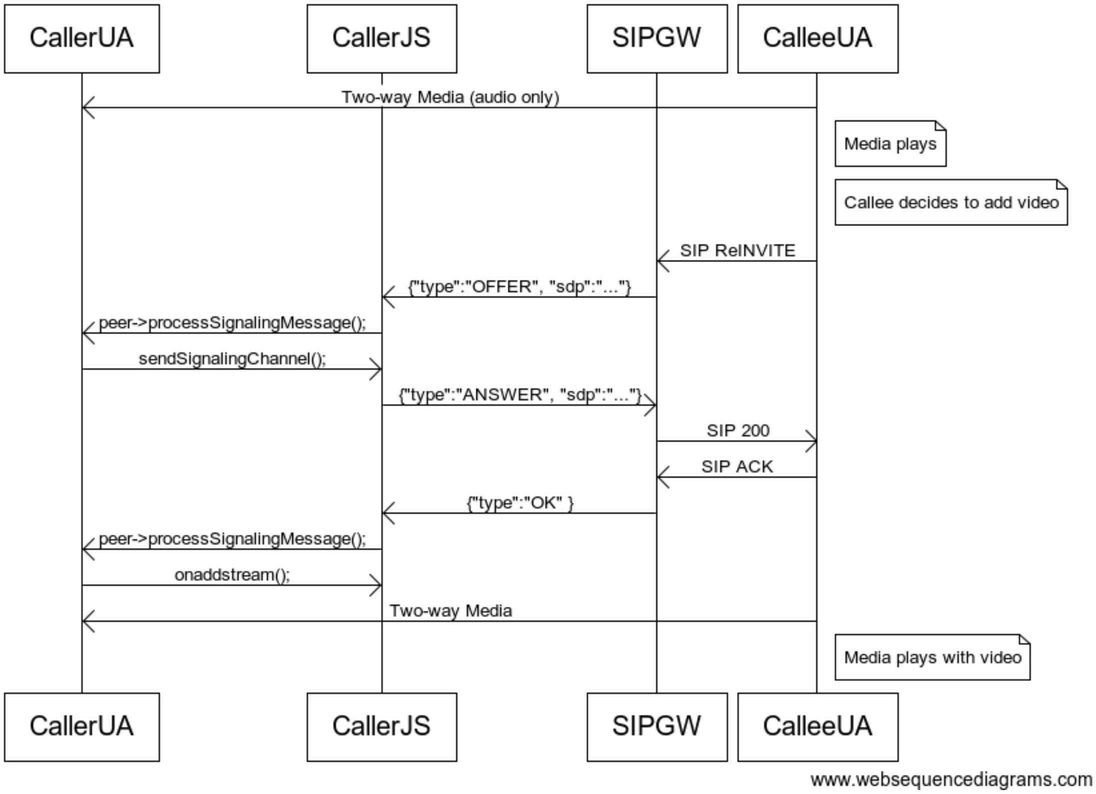
```
 {"type":"OK",
  "offererSessionId":"36707f69b",
  "answererSessionId":"8321234356",
  "seq": 1
 }
```

```
{ACK sip:calleeua@client.sippy.example.com SIP/2.0
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: CallerUA <sip:callerua@atlanta.example.com>;tag=36707f69b
To: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 1 ACK
}
```

TOC

2.2. Add New Media (video)

This scenario describes the message exchanges when CalleeUA decides to add video as media to an existing audio-only session



Message Details

On receipt of SIP:INVITE with SDP for video, the signaling gateway maps SIP:INVITE to ROAP:OFFER(section 5.3.1 of ROAP**[ROAP]**) and sends it to CallerUA indicating the intent.

```
{INVITE sip:callerua@atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
To: CallerUA <sip:callrua@atlanta.example.com>;tag=36707f69b
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 2 INVITE
Contact: <sip:calleeua@client.sippy.example.com;transport=udp>
Content-Type: application/sdp

v=0
o=callerua 1429 0  IN IP4 client.atlanta.example.com
s=SIP Call
c=IN IP4 192.0.2.101
t=0 0
m=video 1024 RTP/AVP 97
a=fmtp:97 profile-level-id=42E00C
a=sendrecv
}
```

CallerUA accepts the new ROAP:OFFER(section 5.3.1 of ROAP**[ROAP]**) and sends ROAP:ANSWER section 5.3.2 of ROAP**[ROAP]**).

```
 {"type":"OFFER",
  "offererSessionId":"36707f69b",
  "answererSessionId":"8321234356",
  "seq": 2,
  "sdp":"
v=0
o=callerua 1429 0  IN IP4 client.atlanta.example.com
s=Call
c=IN IP4 192.0.2.101
```

```
t=0 0
m=video 1024 RTP/AVP 97
a=fmtp:97 profile-level-id=42E00C
a=sendrecv"
 }
```

Which results in the follwing answer.

```
 {"type":"ANSWER",
  "offererSessionId":"36707f69b",
  "answererSessionId":"8321234356",
  "seq": 2,
  "sdp":"
v=0
o=calleeua 2890844527 2890844527 IN IP4 client.sippy.example.com
s=Call
c=IN IP4 192.0.2.201
t=0 0
m=audio 16834 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendrecv"
 }
```

The signaling gateway maps the ROAP:ANSWER to SIP:200 to be sent to the CalleeUA.

```
         {
{SIP/2.0 200 OK
Via: SIP/2.0/UDP client.sippy.example.com:5060;branch=z9hG4bK74bf9
    ;received=192.0.2.201
From: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
To: CallerUA <sip:calleeua@sippy.example.com>;tag=36707f69b
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 102 INVITE
Contact: <sip:callerua@client.atlanta.example.com;transport=udp>
Content-Type: application/sdp

v=0
o=calleeua 2890844527 2890844527 IN IP4 client.sippy.example.com
s=SIP Call
c=IN IP4 192.0.2.201
t=0 0
m=video 1024 RTP/AVP 97
a=fmtp:97 profile-level-id=42E00C
a=sendrecv
}
```

CalleeUA accepts the receipt of SIP:200 by sending SIP:ACK. The signaling gateway SIP:ACK to ROAP:OK (section 5.3.2 of ROAP**[ROAP]**) sends it to CallerUA. This completes adding the new media (video) to the existing session.

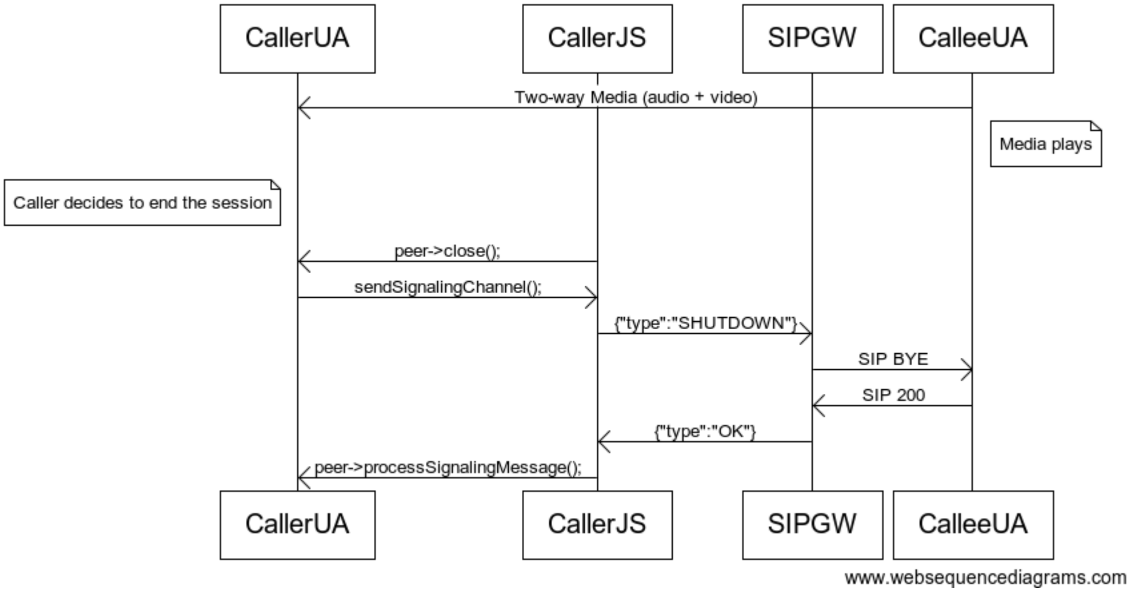
```
{ACK sip:callerua@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: calleeua <sip:calleeua@sippy.example.com>;tag=8321234356
To: callerua <sip:callerua@atlanta.example.com>;tag=36707f69b
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 2 ACK
}
```

```
 {"type":"OK",
  "offererSessionId":"36707f69b",
  "answererSessionId":"8321234356",
  "seq": 2
 }
```

TOC

2.3. SuccessFull Session Ending

This section capture native SIP message descriptions when the caller decides to end the ongoing session.



Message Details

The signaling gateway maps ROAP:SHUTDOWN message from the CallerUA to SIP:BYE and send it to CalleeUA to end the ongoing session.

```
{"type":"SHUTDOWN",
 "offererSessionId":"36707f69b",
 "answererSessionId":"8321234356",
 "seq": 3
}
```

```
{BYE sip:callerua@client.atlanta.example.com SIP/2.0
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: CallerUA <sip:callerua@atlanta.example.com>;tag=36707f69b
To: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 3 BYE
}
```

CalleeUA end's the session from it's side by sending SIP:OK.

```
{SIP/2.0 200 OK
Via: SIP/2.0/UDP client.atlanta.example.com:5060;branch=z9hG4bK74bf9
    ;received=192.0.2.101
From: CallerUA <sip:callerua@atlanta.example.com>;tag=36707f69b
To: CalleeUA <sip:calleeua@sippy.example.com>;tag=8321234356
Call-ID: 00000000-00000003-2331a5b0-2aa0cdf5@atlanta.example.com
CSeq: 3 BYE
Contact: <sip:calleeua@client.sippy.example.com;transport=udp>
}
```

This message SHALL be converted to ROAP:OK(section 5.3.2 of ROAP**[ROAP]**) and sent to caller by the signaling gateway.

```
{"type":"OK",
 "offererSessionId":"36707f69b",
 "answererSessionId":"8321234356",
 "seq": 3
}
```

3. Handling SIP Requests

When the signalling gateway receives a SIP request, the gateway forms the message on the web request side in the following way:

1. The SIP methods INVITE, ACK, BYE, CANCEL are mapped to messageType OFFER, OK, SHUTDOWN, SHUTDOWN respectively
2. The Seq on web side is formed from the numeric portion of CSeq header field value from the SIP side.
3. The offererSessionId is formed by a JSON object string that has an call-id attribute containing the SIP call-id header field value and a from-tag attribute containing the SIP from-tag.
4. If there is a SIP to-tag, it is used for the answererSessionId.
5. If there is a SIP body containing SDP, it is copied into the SDP parameter on web side.
6. The setSessionToken is formed by a JSON object string that has contact attribute that contains the SIP contact header field value and an route attribute which is an array that has the values of the SIP route header field values in reverse order.
7. The setResponseToken formed by a JSON object string that has via attribute that is an array containing the SIP via headers field values. The JSON object also includes an attribute that holds the request method. The gateway MAY include any other SIP headers in an attribute named headers which is an array with one header field in each entry.

TOC

4. Handling SIP Responses

When the signalling gateway receives a SIP response, the gateway forms the message on the web request side in the following way:

1. The SIP responses 180 is mapped to ANSWER with more_coming. A 200 response that contains SDP is mapped to ANSWER. 481 is mapped to NOMATCH. 408 is mapped to TIMEOUT. 486 is mapped to REFUSED. 491 is mapped to CONFLICT. All other SIP 3xx to 6xx responses are mapped to FAILED.
2. The Seq on web side is formed from the numeric portion of CSeq header field value from the SIP side.
3. The offererSessionId is formed by a JSON object string that has an call-id attribute containing the SIP call-id header field value and a from-tag attribute containing the SIP from-tag.
4. The SIP to-tag is used for the answererSessionId.
5. If there is a SIP body containing SDP, it is copied into the SDP parameter on web side.
6. The setSessionToken is formed by a JSON object string that has contact attribute that contains the SIP contact header field value and an route attribute which is an array that has the values of the SIP route header field values.
7. The setResponseToken formed by a JSON object string that has via attribute that is an array containing the SIP via headers field values. The gateway MAY include any other SIP headers in an attribute named headers which is an array with one header field in each entry.

TOC

5. Handling Web Messages

When the signalling gateway receives a WEB message, the gateway forms the message on the SIP side in the following way:

1. The messageType OFFER, ANSWER with more_coming, ANSWER, OK, NOMATCH, TIMEOUT, REFUSED, CONFLICT, FAILED are mapped to INVITE, 180, 200, ACK, 481, 408, 486, 491, 500 respectively.
2. The messageType SHUTDOWN is mapped to a CANCEL if the answererSessionId is empty and to BYE otherwise
3. For SIP responses, The numeric portion of the CSeq is formed by taking the number portion from the Seq field. If the setResponseToken contains a method name, that is used for the method portion of the CSeq otherwise if it does not exist, the request method of the SIP message is used.
4. The Call-ID header field values is formed from the call-id attribute of the offererSessionId.
5. The from-tag is formed from the from-tag attribute of the offererSessionId.
6. If there is a answererSessionId, it is used for the SIP to-tag.
7. If there is a SDP parameter, it is used as a SIP SDP body and the content type of and content length headers are set appropriately.
8. If there is a sessionToken that contains a contact attribute, it is used to form the SIP contact header field value.
9. If there is a sessionToken that contains a route array, it is used to form the SIP route header field values.
10. If there is a responseToken that contains a via array, it is used to form the SIP via header field values.

TOC

6. Handling Glare

SIP uses a timer based mechanism as defined by Section 14.1 and 14.2 of [RFC3261] and further clarified by Section 3.5 of [RFC6141]. This mechanism consists of any UAS sending an 491 response to any INVITE in a SIP dialog when having an outstanding INVITE. Then depending on the UAC relation to the SIP Dialog it will start a timer in the interval 0-2 seconds or 2.1-4 seconds before restarting. As SIP UA is both a UAC and a UAS, this normally results in that the INVITEs in both direction gets terminated when a glare occur.

Depending on the role of the gateway and the SIP end-point the gateway can act in different ways to minimize the delay to handle the glare situation and depending on where it happens. From the gateways perspective the glare situation can be detected between the gateway and the ROAP peer or between the gateway and the SIP UA. Depending on where it occur different actions should be taken.

Glare between a ROAP Peer and a gateway can occur when the gateway have issued an ROAP OFFER based on a translation of an SIP INVITE. When issuing that ROAP OFFER it needs to assign a tiebreaker value. Two reasonable choices exist, either a random value or 4,294,967,295. Using 4,294,967,295 would ensure that the gateway's ROAP Message always wins over a ROAP peer's message. This to avoid incurring SIP resolution mechanism timeouts on the session transaction. The downside would be if two gateway's gets interconnected then this will result in an immediate glare situation that requires retry. Thus using random values is anyway required to be supported. The recommendation is that the gateway uses tie-breaker value 4,294,967,295 until a double glare situation arises or it is known that the peer is a gateway, in which cases it generates a random value.

If the gateway's OFFER wins the gateway responds to the ROAP peer's OFFER with Error:CONFLICT.

If the gateway's OFFER loses the gateway will need to communicate this to the SIP UA. Thus it needs to send a 491 to the SIP UA. Now the gateway has the responsibility to provide the SIP UA with the ROAP Peer's OFFER. That could be accomplished by sending a SIP Invite to the SIP UA after the 491 that indicates that it is a new SIP Invite rather than the one that caused the conflict. How likely that is to succeed depends on the SIP UA's role in establishing the SIP dialog, if it owns the Call-ID of dialog ID its timer T as a result of the 491 response will be in the 2.1-4 interval and likely give the new INVITE message time to be processed. However, if the SIP UA doesn't own Call-ID of the dialog ID, then it will use a timer T that is in the interval 0-2 seconds, thus potentially resulting in a new glare situation.

In case the glare occurs after the gateway translated and sent the ROAP OFFER as a SIP INVITE the gateway will receive an SIP Invite before a response to its SIP Invite. However, the gateway has the possibility to cause the ROAP peers SIP INVITE to be the losing one on the ROAP side of the gateway by translating the SIP UA's INVITE into a ROAP OFFER and the assign it a tiebreaker value greater than the ROAP Peer's value. That way terminating it and letting the SIP side's offer go through immediate rather than being stuck in waiting for a timer to cause a reINVITE. In case the ROAP peer tiebreaker value is 4,294,967,295 the gateway can either respond with a 491 to the SIP UA's INVITE and then send the ROAP Peer's as new message after the 491 response, or send a ROAP OFFER back that causes double glare.

TOC

7. Limitations

The following things, if used on the SIP side, will not interoperate:

- Redirection via 3xx
- UPDATE / PRACK
- REFER
- SIP to pre RFC 3261 devices that don't support to and from tags.
- SUB/NOTify
- SIP INVITES that do not contain an SDP offer
- SIP extensions to RFC 3261.

TOC

8. Security Considerations

TBD

TOC

9. IANA Considerations

This document requires no actions from IANA.

TOC

10. Acknowledgments

The text for the glare section was provided by Magnus Westerlund.

TOC

11. References

TOC

11.1. Normative References

[RFC3264] Rosenberg, J. and H. Schulzrinne, "**An Offer/Answer Model with Session Description Protocol (SDP)**," RFC 3264, June 2002 (**TXT**).

[RFC2119] **Bradner, S.**, "**Key words for use in RFCs to Indicate Requirement Levels**," BCP 14, RFC 2119, March 1997 (**TXT**, **HTML**, **XML**).

TOC

11.2. Informative References

[RFC3261]	Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "**SIP: Session Initiation Protocol**," RFC 3261, June 2002 (**TXT**).
[I-D.ietf-rtcweb-use-cases-and-requirements]	Holmberg, C., Hakansson, S., and G. Eriksson, "**Web Real-Time Communication Use-cases and Requirements**," draft-ietf-rtcweb-use-cases-and-requirements-06 (work in progress), October 2011 (**TXT**).
[ROAP]	Jennings, C. and J. Rosenberg, "**RTCWeb Offer/Answer Protocol (ROAP)**," draft-jennings-rtcweb-signaling (work in progress), October 2011 (**TXT**).

TOC

Authors' Addresses

Cullen Jennings
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
Phone: +1 408 421-9990
Email: **fluffy@cisco.com**

Suhas Nandakumar
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA
Email: **snandaku@cisco.com**

Christer Holmberg
Ericsson
Hirsalantie 11
Jorvas 02420
Finland
Email: **christer.holmberg@ericsson.com**