

**Session Initiation Protocol Service Example -- Music on Hold
draft-worley-service-example-01**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on July 19, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

The "music on hold" feature is one of the most desired features of telephone systems in the business environment. "Music on hold" is when one party to a call has the call "on hold", the other party receives a media stream (often either music or advertising). Architectural features of SIP make it difficult to implement music-on-hold in a way that is fully compliant with the standards. The implementation of music-on-hold described in this document is fully effective and standards-compliant, but is simpler than the methods previously documented.

Table of Contents

1.	Introduction	3
2.	Technique	4
2.1.	Placing a Call on Hold and Providing an External Media Stream	4
2.2.	Taking a Call off Hold and Terminating the External Media Stream	5
2.3.	Example Message Flow	6
2.4.	Managing o= Lines	14
3.	Advantages	16
4.	Security Considerations	17
5.	Acknowledgments	18
6.	Revision History	19
6.1.	Changes from draft-worley-service-example-00 to draft-worley-service-example-01	19
7.	References	20
7.1.	Normative References	20
7.2.	Informative References	20
	Author's Address	21
	Intellectual Property and Copyright Statements	22

1. Introduction

Within SIP[1]-based systems, it is desirable to be able to provide features that are similar to those provided by traditional telephony systems. A frequently requested feature is "music on hold": The music-on-hold feature is when one party to a call has the call "on hold", that party's telephone provides a media stream to be rendered to the other party.

Architectural features of SIP make it difficult to implement music-on-hold in a way that is fully compliant with the standards. The purpose of this document is to describe a method that is reasonably simple yet fully effective and standards-compliant.

2. Technique

The essence of the technique is that when the user's UA (referred to hereafter as the "executing UA") establishes the hold state, it uses third party call control mechanisms to direct RTP from an audio source service to the held party. The executing UA establishes a dialog with the service to negotiate the audio stream, but does not act as a media relay; media flows from the service to the held party.

In order to accomplish this within the offer/answer model, the executing UA sends a re-INVITE to the remote UA to establish the hold state, but in that INVITE it provides no SDP offer, thus compelling the remote UA to provide an SDP offer. The executing UA then uses that offer (modified as described in [Section 2.4](#)) in a new INVITE to the external media source. The external media source is thus directed to provide media directly to the remote UA. The media source's answer SDP is returned to the remote UA in the ACK to the re-INVITE.

2.1. Placing a Call on Hold and Providing an External Media Stream

1. The executing user instructs the executing UA to put the dialog on-hold.
2. The executing UA sends a re-INVITE without SDP to the remote UA, which forces the remote UA to provide an SDP offer in its 2xx response. The Contact header of the re-INVITE SHOULD include the '+sip.rendering="no"' field parameter so that the intent is clear and so that any dialog events indicate that it is putting the call on hold.[\[6\]](#) (The sip.rendering feature parameter is not required by the MOH mechanics, but rather to facilitate reporting the on-hold status in dialog events.)
3. The remote UA sends a 2xx to the re-INVITE, and includes an SDP offer giving its own listening address/port. If the remote UA understands the sip.rendering feature parameter, the offer may indicate that it will not send media by specifying the media directionalities as "recvonly" (the reverse of "on-hold") or perhaps "inactive". But the remote UA may offer to send media.
4. The executing UA uses this offer to construct the offer SDP of an initial INVITE that it sends to the configured music-on-hold (MOH) source. The SDP in this request is, with the exceptions below, copied from the SDP returned by the remote UA in the previous step, particularly regarding the provided listening address/port and payload type numbers. The changes to the SDP offer are:

- * The o= line of the offer is modified according to [Section 2.4](#).
 - * Any explicit or implicit media directionality of "sendrecv" is replaced with "recvonly", and any media directionality of "sendonly" is replaced with "inactive".
5. The MOH sends a 2xx response to the INVITE, which contains an SDP answer that should include its media source address as its listening address/port. This SDP must necessarily[2] specify "sendonly" or "inactive" as the directionality for all media streams. (Although this address/port should receive no RTP, the answer SDP will reach the remote UA, informing it of the address/port from which the MOH media will come, and presumably preventing it from ignoring the MOH media as SPIT. This requires the SDP answer to provide a real address/port in the c= line, even though the MOH server does not receive RTP.)
 6. The executing UA copies the MOH service SDP answer, modifying it as listed below, and sends it as its SDP answer in the ACK for the re-INVITE to the remote UA.
 - * The o= line of the answer is modified according to [Section 2.4](#).
 7. Due to the sip.rendering feature parameter in the Contact of the re-INVITE and the media directionality in the SDP answer contained in the ACK, the on-hold state (at the executing end) of the dialog is established. (The sip.rendering feature parameter will be reported in the dialog events generated by both UAs, allowing any monitoring application to determine that the call is on hold.)
 8. After this point, the MOH source generates RTP containing the music-on-hold media, and sends it directly to the listening address/port of the remote UA. The executing UA maintains two dialogs (one to the remote UA, one to the MOH source), but does not see or handle the MOH RTP.

[2.2](#). Taking a Call off Hold and Terminating the External Media Stream

1. The executing user instructs the executing UA to take the dialog off-hold.
2. The executing UA sends a re-INVITE to the remote UA with SDP that requests to receive media. The Contact header of the re-INVITE does not include the '+sip.rendering="no"' field parameter. (It may contain a sip.rendering field parameter with value "yes" or "unknown", or it may omit the field parameter.) Thus this INVITE

removes the on-hold state (at the executing end) of the dialog.

3. When the remote UA sends a 2xx response to the re-INVITE, the executing UA sends a BYE request in the dialog to the MOH source to terminate the MOH media stream.
4. After this point, the MOH source does not generate RTP and ordinary RTP flow is re-established in the original dialog.

2.3. Example Message Flow

This section shows a message flow which is an example of this technique. The scenario is: Alice establishes a call with Bob. Bob then places the call on hold, with music-on-hold provided from an external server. Bob then takes the call off hold.

Note that this is just one possible message flow that illustrates this technique; numerous variations on these operations are allowed by the applicable standards.

Alice Bob Music Server

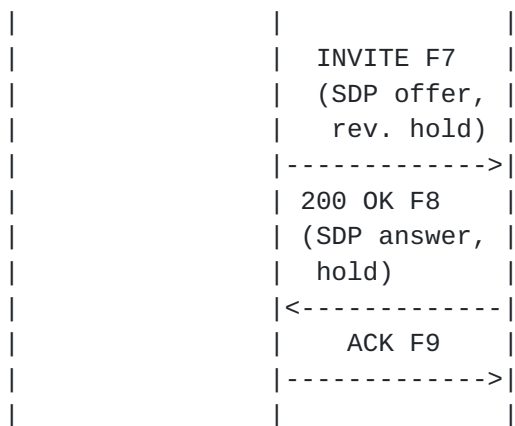
Alice establishes the call:

	INVITE F1			
	----->			
	180 Ringing F2			
	<-----			
	200 OK F3			
	<-----			
	ACK F4			
	----->			
	RTP			
	<=====			

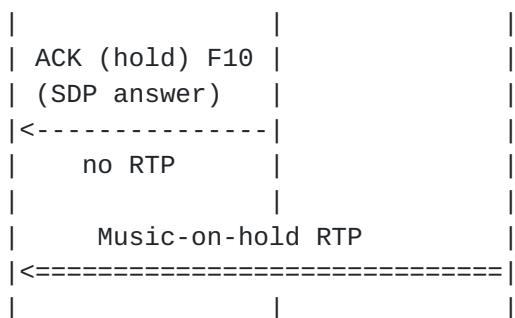
Bob places Alice on hold, compelling Alice's UA to provide SDP:

	INVITE F5			
	(no SDP)			
	<-----			
	200 OK F6			
	(SDP offer)			
	----->			

Bob's UA initiates music-on-hold:

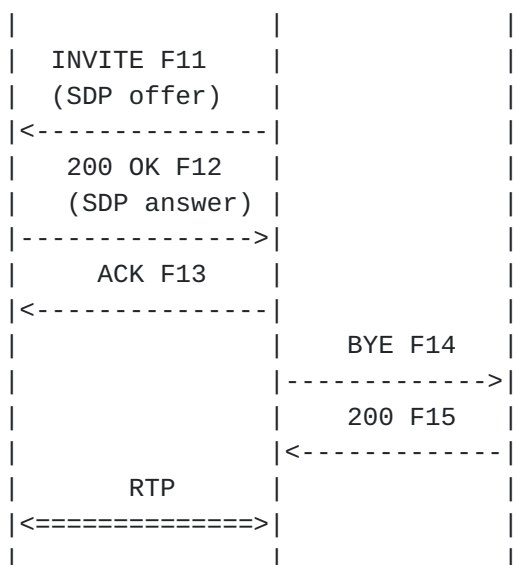


Bob's UA provides SDP answer containing the address/port of the Music Server:



The music on hold is active.

Bob takes Alice off hold:



The normal media session between Alice and Bob is resumed.

Message Details

/* Alice calls Bob. */

F1 INVITE Alice -> Bob

```
INVITE sips:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TLS atlanta.example.com:5061
    ;branch=z9hG4bK74bf9
Max-Forwards: 70
From: Alice <sips:alice@atlanta.example.com>;tag=1234567
To: Bob <sips:bob@biloxi.example.com>
Call-ID: 12345600@atlanta.example.com
CSeq: 1 INVITE
Contact: <sips:a8342043f@atlanta.example.com;gr>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu
Content-Type: application/sdp
Content-Length: [omitted]
```

```
v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

F2 180 Ringing Bob -> Alice

```
SIP/2.0 180 Ringing
Via: SIP/2.0/TLS atlanta.example.com:5061
    ;branch=z9hG4bK74bf9
    ;received=192.0.2.103
From: Alice <sips:alice@atlanta.example.com>;tag=1234567
To: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 1 INVITE
Contact: <sips:bob@biloxi.example.com>
Content-Length: 0
```

F3 200 OK Bob -> Alice

SIP/2.0 200 OK

Via: SIP/2.0/TLS atlanta.example.com:5061
;branch=z9hG4bK74bf9
;received=192.0.2.103
From: Alice <sips:alice@atlanta.example.com>;tag=1234567
To: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 1 INVITE
Contact: <sips:bob@biloxi.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces
Content-Type: application/sdp
Content-Length: [omitted]

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

F4 ACK Alice -> Bob

ACK sips:bob@biloxi.example.com SIP/2.0
Via: SIP/2.0/TLS atlanta.example.com:5061
;branch=z9hG4bK74bfd
Max-Forwards: 70
From: Alice <sips:alice@atlanta.example.com>;tag=1234567
To: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 1 ACK
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces
Content-Length: 0

/* Bob places Alice on hold. */

/* The re-INVITE contains no SDP, thus compelling Alice's UA
to provide an offer. */

F5 INVITE Bob -> Alice

INVITE sips:a8342043f@atlanta.example.com;gr SIP/2.0
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bK874bk
To: Alice <sips:alice@atlanta.example.com>;tag=1234567

From: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 712 INVITE
Contact: <sips:bob@biloxi.example.com>;+sip.rendering="no"
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces
Content-Length: 0

/* Alice's UA provides an SDP offer.
Since it does not know that it is being put on hold,
the offer is the same as the original offer and describes
bidirectional media. */

F6 200 OK Alice -> Bob

SIP/2.0 200 OK
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bK874bk
;received=192.0.2.105
To: Alice <sips:alice@atlanta.example.com>;tag=1234567
From: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 712 INVITE
Contact: <sips:a8342043f@atlanta.example.com;gr>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu
Content-Type: application/sdp
Content-Length: [omitted]

v=0
o=alice 2890844526 2890844526 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=active

/* Bob's UA initiates music-on-hold. */

/* This INVITE contains Alice's offer, but with the media
direction set to "reverse hold", receive-only. */

F7 INVITE Bob -> Music Server

INVITE sips:music@server.example.com SIP/2.0
Via: SIP/2.0/TLS biloxi.example.com:5061

;branch=z9hG4bKnashds9
Max-Forwards: 70
From: Bob <sips:bob@biloxi.example.com>;tag=02134
To: Music Server <sips:music@server.example.com>
Call-ID: 4802029847@biloxi.example.com
CSeq: 1 INVITE
Contact: <sips:bob@biloxi.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu
Content-Type: application/sdp
Content-Length: [omitted]

v=0
o=bob 2890844534 2890844534 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=recvnly

F8 200 OK Music Server -> Bob

SIP/2.0 200 OK
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bKnashds9
;received=192.0.2.105
From: Bob <sips:bob@biloxi.example.com>;tag=02134
To: Music Server <sips:music@server.example.com>;tag=56323
Call-ID: 4802029847@biloxi.example.com
Contact: <sips:music@server.example.com>
CSeq: 1 INVITE
Content-Length: [omitted]

v=0
o=MusicServer 2890844576 2890844576 IN IP4 server.example.com
s=
c=IN IP4 server.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly

F9 ACK Bob -> Music Server

ACK sips:music@server.example.com SIP/2.0

Via: SIP/2.0/TLS server.example.com:5061
;branch=z9hG4bK74bT6
From: Bob <sips:bob@biloxi.example.com>;tag=02134
To: Music Server <sips:music@server.example.com>;tag=56323
Max-Forwards: 70
Call-ID: 4802029847@biloxi.example.com
CSeq: 1 ACK
Content-Length: 0

/* Bob's UA now sends the ACK that completes the re-INVITE
to Alice and completes the SDP offer/answer.
The ACK contains the SDP received from the Music Server,
and thus contains the address/port from which the Music Server
will send media. */

F10 ACK Bob -> Alice

ACK sips:a8342043f@atlanta.example.com;gr SIP/2.0
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bKq874b
To: Alice <sips:alice@atlanta.example.com>;tag=1234567
From: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 712 ACK
Contact: <sips:bob@biloxi.example.com>;+sip.rendering="no"
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces
Content-Length: [omitted]

v=0
o=bob 2890844527 2890844528 IN IP4 biloxi.example.com
s=
c=IN IP4 server.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=sendonly

/* Bob picks up the call by sending a re-INVITE to Alice. */

F11 INVITE Bob -> Alice

INVITE sips:a8342043f@atlanta.example.com;gr SIP/2.0
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bK874bk
To: Alice <sips:alice@atlanta.example.com>;tag=1234567
From: Bob <sips:bob@biloxi.example.com>;tag=23431

Call-ID: 12345600@atlanta.example.com
CSeq: 713 INVITE
Contact: <sips:bob@biloxi.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces
Content-Type: application/sdp
Content-Length: [omitted]

v=0
o=bob 2890844527 2890844529 IN IP4 biloxi.example.com
s=
c=IN IP4 biloxi.example.com
t=0 0
m=audio 3456 RTP/AVP 0
a=rtpmap:0 PCMU/8000

F12 200 OK Alice -> Bob

SIP/2.0 200 OK
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bK874bk
;received=192.0.2.105
To: Alice <sips:alice@atlanta.example.com>;tag=1234567
From: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 713 INVITE
Contact: <sips:a8342043f@atlanta.example.com;gr>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu
Content-Type: application/sdp
Content-Length: [omitted]

v=0
o=alice 2890844526 2890844527 IN IP4 atlanta.example.com
s=
c=IN IP4 atlanta.example.com
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000

F13 ACK Bob -> Alice

ACK sips:a8342043f@atlanta.example.com;gr SIP/2.0
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bKq874b
To: Alice <sips:alice@atlanta.example.com>;tag=1234567

From: Bob <sips:bob@biloxi.example.com>;tag=23431
Call-ID: 12345600@atlanta.example.com
CSeq: 713 ACK
Contact: <sips:bob@biloxi.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces
Content-Length: 0

F14 BYE Bob -> Music Server

BYE sips:music@server.example.com SIP/2.0
Via: SIP/2.0/TLS biloxi.example.com:5061
;branch=z9hG4bK74rf
Max-Forwards: 70
From: Bob <sips:bob@biloxi.example.com>;tag=02134
To: Music Server <sips:music@server.example.com>;tag=56323
Call-ID: 4802029847@biloxi.example.com
CSeq: 2 BYE
Contact: <sips:bob@biloxi.example.com>
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY
Supported: replaces, gruu
Content-Length: [omitted]

F15 200 OK Music Server -> Alice

SIP/2.0 200 OK
Via: SIP/2.0/TLS atlanta.example.com:5061
;branch=z9hG4bK74rf
;received=192.0.2.103
From: Bob <sips:bob@biloxi.example.com>;tag=02134
To: Music Server <sips:music@server.example.com>;tag=56323
Call-ID: 4802029847@biloxi.example.com
CSeq: 2 BYE
Content-Length: 0

/* Normal media session between Alice and Bob is resumed */

2.4. Managing o= Lines

The rules [Section 2.4](#) regarding the use of o= lines in successive SDP offers/answers during a dialog are quit strict. In particular, successive SDPs sent by one UA in the dialog must have identical o= lines, other than that the version number field must be incremented by 1. The single exception is that if two or more successive SDPs are exactly the same, they may have the same version number.

In order for the executing UA to conform to these rules, it must modify the o= lines in any SDP that it passes from one dialog to the other. In particular:

1. The SDP in the initial INVITE to to the MOH server must have its o= line replaced with a newly constructed o= line that is valid within the dialog with the MOH server -- See [\[3\]](#) for the details.
2. Within any later SDP copied from the original dialog to the MOH dialog, the o= line must be replaced with a copy of the o= line that the executing UA used in the previous SDP that it sent in the MOH dialog, with the version number incremented by 1.
3. Within any SDP copied from the MOH dialog to the original dialog, the o= line must be replaced with a copy of the o= line that the executing UA used in the previous SDP that it sent in the original dialog, with the version number incremented by 1.
4. When the executing UA takes the call off-hold, the o= line of the SDP that it sends in the re-INVITE must have a version number one greater than in the o= line of the last SDP the executing UA copied from the MOH dialog.

3. Advantages

This technique for providing music-on-hold has advantages over other methods now in use:

1. The original dialog is not transferred to another UA, so the "remote endpoint URI" displayed by the remote endpoint's user interface and dialog event package[6] does not change during the call.[\[4\]](#)
2. The music-on-hold media is sent directly from the music-on-hold source to the remote UA, rather than being relayed through the holding UA.
3. The remote UA sees, in the incoming SDP, the address/port that the MOH server will send MOH media from, thus allowing it to render the media, even if it is filtering incoming media based on originating address as a SPIT preventative.
4. The technique requires relatively simple manipulation of SDP, and in particular: (1) does not require a SIP agent to modify unrelated SDP to be acceptable to be sent within an already established sequence of SDP (a problem with [\[5\]](#)), and (2) does not require converting an SDP answer into an SDP offer (which was a problem with the -00 version of this document, as well as with [\[5\]](#)).

4. Security Considerations

Some UAs filter incoming media based on the address of origin in order to avoid SPIT. This technique ensures that any UA that should render MOH media will be informed of the source address via the SDP that it receives. This should allow such UAs to filter without interfering with MOH operation.

5. Acknowledgments

The original version of this proposal was derived from [\[5\]](#) and the similar implementation of MOH in the Snom UA. Significant improvements to sequence of operations, allowing improvements to the SDP handling, were suggested by Venkatesh[7].

John Elwell[8] pointed out the need for the executing UA to pass through re-INVITES/UPDATES in order to allow ICE negotiation, which suggested to me the need for pass-through to handle the remote UA placing its end of the call on-hold.

This version benefitted from Scott Lawrence's careful reading and comments.

6. Revision History

6.1. Changes from [draft-worley-service-example-00](#) to [draft-worley-service-example-01](#)

Removed the original "Example Message Flow" and promoted the "Alternative Example Message Flow" to replace it because of a number of flaws that were described in the discussion of -00 on the SIPPING mailing list.

Described the use of the sip.rendering feature parameter to indicate on-hold status.

Added Acknowledgments section.

Added separate section on the management of o= lines.

7. References

7.1. Normative References

- [1] 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.
- [2] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with the Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [3] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.

7.2. Informative References

- [4] Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, "Session Initiation Protocol Service Examples", I-D [draft-ietf-sipping-service-examples-13](#), July 2007.
- [5] Johnston, A., Sparks, R., Cunningham, C., Donovan, S., and K. Summers, "Session Initiation Protocol Service Examples", I-D [draft-ietf-sipping-service-examples-11](#), October 2006.
- [6] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", [RFC 4235](#), November 2005.
- [7] Venkatesh, "Subject: Re: [Sipping] I-D ACTION:[draft-ietf-sipping-service-examples-11.txt](#)", IETF Sipping mailing list msg12180, October 2006.
- [8] Elwell, J., "Subject: [Sipping] RE: I-D Action:[draft-worley-service-example-00.txt](#)", IETF Sipping mailing list msg14678, November 2007.

Author's Address

Dale R. Worley
Bluesocket Inc.
10 North Ave.
Burlington, MA 01803
US

Phone: +1 781 229 0533 x173

Email: dworley@pingtel.com

URI: <http://www.pingtel.com>

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The IETF Trust (2008). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

