An Extension to the Session Initiation Protocol (SIP) for Media Loopback
draft-hedayat-media-loopback-00

Status of this Memo

By submitting this Internet-Draft, we certify that any applicable patent or other IPR claims of which we are aware have been disclosed and any of which we become aware will be disclosed, in accordance with RFC 3668 (BCP 79).

By submitting this Internet-Draft, we accept the provisions of Section 3 of RFC 3667 (BCP 78).

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.
Abstract

The wide deployment of SIP-based VoIP services has introduced new challenges in managing and maintaining voice quality, reliability, and overall performance. In particular, media delivery is an area that needs attention. One method of meeting these challenges is monitoring the overall service by establishing calls where the media is looped back by the answering entity to the calling entity. This is typically referred to as "active monitoring" of services. Such calls are especially popular in ensuring the quality of transport to the edge of a given VoIP service. Today in SIP-based networks, short of running 'ping' and 'traceroute' to the edge, service providers are left without the necessary tools to actively monitor, manage, and diagnose quality issues with their service. The extension defined herein adds new SDP media attributes which enables SIP User Agent Clients to initiate calls wherein the media is looped back by the SIP User Agent Servers thereby enabling the User Agent Client to proactively measure the media performance of the network.

Table of Contents

1. Introduction.................................................. 3
2. Terminology................................................... 3
3. User Agent Client Behavior.................................... 4
4. User Agent Server Behavior.................................... 4
5. Proxy Behavior................................................ 5
6. SDP Constructs Syntax........................................ 5
   6.1 Loopback Type Attribute................................... 5
   6.2 Loopback Mode Attribute................................... 5
   6.3 Generating the Offer for Loopback Session............... 6
   6.4 Generating the Answer for Loopback Session............... 6
   6.5 Offerer Processing of the Answer.......................... 7
   6.6 Modifying the Session.................................... 7
   6.7 Modifying a Media Stream.................................. 8
7. RTP Requirements.............................................. 8
8. RTCP Requirements............................................. 8
9. Examples........................................................ 9
   9.1 INVITE request with loopback media........................ 9
   9.2 Response to INVITE request accepting loopback media..... 9
   9.3 Response to INVITE request rejecting loopback media..... 9
   9.4 Response to OPTIONS request............................... 10
10. Implementer Guidelines...................................... 10
    10.1 User Agent Client Guidelines............................ 10
    10.2 User Agent Server Guidelines............................. 11
11. Security Considerations..................................... 11
1. Introduction

The overall quality, reliability, and performance of a VoIP service relies on the performance and quality of the media path. These components include the network for transporting signaling and media, and network services such as DNS, and SIP proxy and location servers. One method of monitoring and managing the overall quality of VoIP Services is through monitoring the quality of the media in an active call. This type of "active monitoring" of services is a method of pro-actively managing the performance and quality of VoIP based services.

The goal of active monitoring is to measure the media quality of a VoIP call. A way to achieve this goal is to request an endpoint to loop media back to the other endpoint and to provide media statistics (e.g., RTCP and RTCP XR information). Another method involves deployment of special endpoints that always loop media back for incoming calls. Although the latter method has been used and is functional, it does not scale to support large networks and introduces new network management challenges. Further, it does not offer the granularity of testing a specific endpoint that may be exhibiting problems.

The extension defined in this memo introduces new SDP media attributes that enable SIP clients to initiate calls proposing to loop media back to the calling client. The offer/answer model per RFC 3264 [RFC3264] is used to establish a loopback connection. Furthermore, this extension provides guidelines on handling RTP (RFC 3550) [RFC3550], as well as usage of RTCP (RFC 3550) [RFC3550] and RTCP XR (RFC 3611) [RFC3611] for reporting media related measurements.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [RFC3264] and indicate requirement levels for compliant implementations.
3. User Agent Client Behavior

The client behavior specified here affects the transport processing defined in Section 18.1 of RFC 3261 [RFC3261].

A client compliant to this memo and attempting to establish a call with media loopback MUST include "loopback" media attributes for each individual media description in the offer message. The calling client MUST look for the "loopback" media attributes in the media description(s) of the response from the called User Agent Server (UAS) for confirmation that the request is accepted.

Prior to establishing the call, the client MAY inquire about the capability to accept media loopback requests by sending an OPTIONS request to obtain the description of media capabilities of a UAS per section 11.1 of (RFC 3261) [RFC3261].

4. User Agent Server Behavior

The server behavior specified here affects the transport processing defined in Section 18.2 of (RFC 3261) [RFC3261].

A server compliant to this specification and receiving an offer containing media descriptions with the "loopback" media attributes, MUST acknowledge the request by including the received "loopback" media attributes for each media description in its response. The server MAY reject the "loopback" request for specific media types as defined in section 6.4.1 of this specification.

A server that is not compliant to this specification and which receives an offer with the "loopback" media attributes MAY safely ignore the attribute and treat the incoming offer as a normal request and proceed as specified in Section 18.2 of RFC 3261 [RFC3261].

A server compliant to this specification and receiving an OPTIONS request MUST specify loopback attributes for all media types with loopback support. Furthermore, loopback attributes for all loopback types supported must be specified. Section 9.4 provides an example.
5. Proxy Behavior

Proxies SHOULD propagate the loopback media attributes and MUST NOT modify the attributes.

6. SDP Constructs Syntax

Two new media attributes are defined: one indicates the type of loopback and one indicates the mode of the loopback.

6.1 Loopback Type Attribute

The syntax of the loopback type media attribute is:

\[
a=\text{loopback:<loopback-type>}
\]

The loopback type is used to indicate the type of loopback. The loopback-type values are rtp-pkt-loopback and rtp-media-loopback.

rtp-pkt-loopback: In this mode, the RTP packets are looped back to the sender at a point before the encoder/decoder function in the receive direction to a point after the encoder/decoder function in the send direction. This effectively re-encapsulates the RTP payload with the RTP/UDP/IP overheads appropriate for sending it in the reverse direction. Any type of encoding related functions, such as packet loss concealment, MUST NOT be part of this type of loopback path.

rtp-media-loopback: This loopback is activated as close as possible to the analog interface and after the decoder so that the RTP packets are decoded and subsequently re-encoded prior to transmission back to the sender.

6.2 Loopback Mode Attribute

The loopback mode attribute is used to indicate the mode of the loopback. These attributes can be viewed as additional mode attributes similar to sendonly, recvonly, etc. The syntax of the loopback mode media attribute is:

\[
a=<\text{loopback-mode}>
\]
The loopback-mode values are loopback-source and loopback-mirror.

loopback-source: This attribute specifies that the sender is the media source and expects the receiver to act as a loopback-mirror.

loopback-mirror: This attribute specifies that the receiver will mirror (echo) all received media back to the sender of the RTP stream. No media is generated locally by the reciever for transmission in the mirrored stream.

6.3 Generating the Offer for Loopback Session

If an offerer wishes to make a loopback request, it MUST include both the loopback-type and loopback-mode attribute in a valid SDP offer:

Example:

a=loopback-type:rtp-media-loopback
a=loopback-source

Note: A loopback offer in a given media description MUST NOT contain the standard mode attributes sendonly, recvonly, sendrecv or inactive.

For loopback-source media (e.g. audio) streams, the port number and the address in the offer indicates where the offerer would like to receive the media stream. The payload type numbers indicate the value of the payload the offerer expects to receive, and would prefer to send. However, the answer might indicate a different payload type number for the same codec. In that case, the offerer MUST send the payload type received in the answer.

The loopback-source attribute can only appear in an offer; if received in an answer, it SHOULD be ignored.

6.4 Generating the Answer for Loopback Session

If a stream is offered with loopback-source, the corresponding stream MUST be loopback-mirror, provided that answerer is capable of supporting the requested loopback-type. It is also recommended that the loopback-type received in the offer is also echoed in the answer.

For example, if the offer contains:
The answer that is capable of supporting the offer MUST contain:

- `a=loopback-type:rtp-media-loopback`
- `a=loopback-source`

The loopback-mirror attribute can only appear in an answer; if received in an offer, it SHOULD be ignored.

### 6.4.1 Rejecting the Loopback Offer

An offered stream with `loopback-source` MAY be rejected, if the loopback-type is not specified, the specified loopback-type is not supported, or the endpoint cannot honor the offer for any other reason. The Loopback request may be rejected by setting the media port number to zero, according to RFC 3264 [RFC3264], in the answer.

### 6.5 Offerer Processing of the Answer

For standard offerer processing of the answer, refer to section 7 of RFC 3264 [RFC3264]. The answer to a loopback-source MUST be loopback-mirror. In addition, the "m=" line MUST contain at least one code that the answerer is willing to both send and receive. In cases where the offer contains the loopback-mirror attribute, as defined in section 6.4, the answer MUST contain the loopback attributes as they were present in the original offer.

If the answer does not contain `a=loopback-mirror` or contains any other standard mode attributes, it is assumed that the loopback extensions are not supported by the target UA. In such cases the offerer may send a BYE to terminate the session.

### 6.6 Modifying the Session

At any point during the loopback session, either participant may issue a new offer to modify the characteristics of the previous session, as defined in section 8 of RFC 3264 [RFC3264]. This also includes transitioning from a normal media processing mode to loopback mode, and vice versa.
6.7 Modifying a Media Stream

Characteristics of a media stream can be modified as defined in section 8 of RFC 3264 [RFC3264].

7. RTP Requirements

A UAS that is compliant to this specification and accepting a media with rtp-pkt-loopback loopback-type MUST re-generate all of the RTP header fields as it does when transmitting other media. However, the UAS MUST maintain the timing information of the received RTP packets when generating the RTP timestamp for the transmit packets. Maintaining the timing information of the RTP packets enables the offerer to re-construct the incoming media and take account for impairments from gaps in the media due to packet loss. Note that RTP Sequence numbers are re-generated by the UAS and will not provide packet loss information to the receiver of the loopback media.

A UAS that is compliant to this specification and accepting a media with rtp-media-loopback loopback-type MUST transmit all received media back to the sender. The incoming media MUST be treated as if it were to be played (e.g. the media stream MAY receive treatment from PLC algorithms). The UAS MUST re-generate all the RTP header fields as it would when transmitting media. The UAS MAY choose to encode the loopback media according to any of the media descriptions supported by the UAC. Furthermore, in cases where the same media type is looped back, the UAS MAY choose to preserve number of frames/packet and bitrate of the encoded media according to the received media.

8. RTCP Requirements

The use of the loopback attribute is intended for monitoring of media quality of the call. Consequently the media performance information must be exchanged between the client and the server. A UAS or UAC that is compliant to this specification MUST support RTCP per [RFC3550] and RTCP-XR per RFC 3611 [RFC3611]. Furthermore, both the client and the server MUST support RTCP-XR Statistics Summary Report Block and VoIP Metric Reports Block per sections 4.6 and 4.7 of RFC 3611 [RFC3611]. The client and the
server MAY support other RTCP-XR reporting blocks as defined by RFC 3611 [RFC3611].

9. Examples

This section provides examples for media descriptions using SDP for different scenarios. The examples are abbreviated and do not show the complete signaling for convenience.

9.1 INVITE request with loopback media

A client sends an INVITE request with SDP which looks like:

v=0
o=user1 2890844526 2890842807 IN IP4 126.16.64.4
s=Example
i=An example session
e=user@example.com
c=IN IP4 224.2.17.12/127
t=0 0
m=audio 49170 RTP/AVP 0
a=loopback:rtp-media-loopback
a=loopback-source

9.2 Response to INVITE request accepting loopback media

A server sends a response with SDP which looks like:

v=0
o=user1 2890844526 2890842807 IN IP4 126.16.64.4
s=Example
i=An example session
e=user@example.com
c=IN IP4 224.2.17.12/127
t=0 0
m=audio 49170 RTP/AVP 0
a=loopback:rtp-media-loopback
a=loopback-mirror

9.3 Response to INVITE request rejecting loopback media
A server sends a response with SDP which looks like:

v=0
o=user1 2890844526 2890842807 IN IP4 126.16.64.4
s=Example
i=An example session
e=user@example.com
c=IN IP4 224.2.17.12/127
t=0 0
m=audio 0 RTP/AVP 0
a=loopback:rtp-media-loopback
a=loopback-mirror

NOTE: Loopback request may be rejected by either not including the loopback mode attribute (for backward compatibility) or setting the media port number to zero, or both, in the response.

9.4 Response to OPTIONS request

A server sends a response with SDP which looks like:

v=0
o=user1 2890844526 2890842807 IN IP4 126.16.64.4
s=Example
i=An example session
e=user@example.com
c=IN IP4 224.2.17.12/127
t=0 0
m=audio 0 RTP/AVP 0
a=loopback:rtp-media-loopback
a=loopback:rtp-pkt-loopback

10. Implementer Guidelines

This section provides guidelines to the implementers of this extension.

10.1 User Agent Client Guidelines

If a client, compliant to this specification and attempting to establish a call with media loopback, finds corresponding media description without the "loopback" media description, it SHOULD terminate the call. Furthermore, the client SHOULD enforce an
upper limit on the duration of the "loopback" media. The recommended values is 60 seconds based on amount of time needed to gather meaningful statistics for a call.

10.2 User Agent Server Guidelines

A server, compliant to this specification SHOULD apply higher priority to non-loopback media modes. As an example, if a server capable of processing only one stream receives an INVITE that contains no loopback request, while it is engaged in a dialog with loopback media, it SHOULD terminate the dialog in which it is engaged and service the new incoming INVITE.

Furthermore, it is RECOMMENDED that the user is not alerted for calls that only have loopback media modes and when such calls are answered automatically.

11. Security Considerations

The security considerations of [RFC3261] apply. Furthermore, given that it is recommended that the user is not alerted for calls that only have loopback media modes the UAS SHOULD be aware of possible denial of service attacks. Therefore, it is recommended that loopback calls are authenticated and the frequency of such calls are limited by the UAS by rejecting the calls.

12. IANA Considerations

There are no IANA considerations associated with this specification.

13. Acknowledgements

The authors wish to thank Flemming Andreasen, Jeff Bernstein, Paul Kyzivat, and Dave Oran for their comments and suggestions.
14. References

14.1 Normative References


Authors’ Addresses

Kaynam Hedayat
Brix Networks
285 Mill Road
Chelmsford, MA  01824
US

Phone: +1 978 367 5611
EMail: khedayat@brixnet.com
URI:  http://www.brixnet.com/

Paul E. Jones
Cisco Systems, Inc.
7025 Kit Creek Rd.
Research Triangle Park, NC  27709
US

Phone: +1 919 392 6948
IPR Disclosure Acknowledgement

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.