

Individual
Internet-Draft

Steven Whitehead
Verizon
Marie-Jose Montpetit
Motorola Connected Home Solutions
Xavier Marjou
France Telecom

Expires: April 30, 2006

February 2006

**An Evaluation of Session Initiation Protocol (SIP)
for use in Streaming Media Applications
draft-whitehead-sip-for-streaming-media-00.txt**

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.

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.

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 April 30, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This draft reviews requirements and evaluates solutions to improve the integration of the Session Initiation Protocol (SIP) [[RFC3261](#)] to the Real Time Streaming Protocol (RTSP and RTSP v2) [RFC 2326 and IDRTSP] especially in the context of converged media services. The document develops a rationale and solution space for using SIP with streaming media applications. This is done by first considering use cases that are used to define generic requirements to help compare the current and potentially new solutions. The range of solutions is sketched out (at high-level) and advantages and disadvantages discussed. The solutions evaluated address different levels of integration from SIP-only approaches to dual-protocol solutions.

Table of Contents

1.	Introduction	X
2.	Terminology.	X
3.	Use Case Scenarios	X
4.	Requirements	X
5.	Rationale for considering SIP.	X
6.	Solution Options	X
6.1	SIP-only Options.	X
6.2	Dual Protocol Options	X
7.	Analysis	X
8.	Recommendations.	X
9.	IANA Considerations	XX
10.	Security Considerations	XX
11.	Acknowledgements	XX
12.	Change History	XX
13.	References	XX
13.1	Normative References	XX
13.2	Informative References	XX
	Author's Address	XX
	Intellectual Property and Copyright Statements	XX

Whitehead

Expires April 30, 2006

[Page 2]

1. Introduction

IP-based networks are continually improving in terms of bandwidth capacity and transport quality of service. At the same time, broadband services are continually expanding globally -- both in terms of reach and value-added services. These developments are leading to an increase in the number and variety of deployment scenarios for streaming media applications. Many of these scenarios impose challenging new requirements on the signaling protocols used by streaming media applications in terms of flexibility, scalability and network independence.

Historically, RTSP [[RFC2326](#)] has been the signaling protocol of choice for streaming media applications. An obvious approach then is to extend RTSP as needed by the new services. This strategy appears able to address many of the new requirements of converged services where multiple heterogeneous streams are merged. Others are very difficult to achieve using current media control (. Moreover, extending RTSP to meet some of these requirements would involve introducing protocol mechanisms already existing elsewhere, namely in SIP and its associated extensions.

An alternative approach is to consider the possibility of using some SIP [[RFC3261](#)] features in streaming media applications. While historically SIP has been used for communication services, the protocol itself is flexible enough (by design) to signal a wide variety of media streams. Moreover, driven in large part by the requirements associated with IP-based communication services, SIP has been extended over the years to address many of the same requirements currently facing next-generation media streaming applications. Rather than reinvent or duplicate protocol mechanisms in RTSP that already exist in SIP, a reasonable strategy may be to find a way to use SIP instead.

This document explores the possibility of using SIP as a signaling protocol for streaming media applications..

The rationale for using SIP is developed by first considering a variety of streaming media use case scenarios. These scenarios are used to derive requirements on the signaling protocol. Given these requirements, the rationale for using SIP is developed by considering SIP's ability to address these requirements. Then, high-level possible strategies for using SIP in streaming media applications are defined.

The document then considers possible ways in which SIP could be used in streaming media applications. Two broad classes of solution strategies are identified: A SIP-only class of solutions whereby SIP is used for both session negotiation and media stream control and a dual-protocol class of solutions whereby SIP is used for session negotiation and another protocol (e.g., RTSP) is used for media stream control. A high-level description of each approach is provided, along with an assessment of its advantages and disadvantages.

Whitehead

Expires April 30, 2006

[Page 3]

The document concludes by summarizing the advantages and disadvantages of each approach and by identifying the approach(es) that most warrant(s) further developments.

Note: The purpose of this document is not to propose in detail one specific approach to using SIP for streaming-media applications. Rather, our intent is to stimulate discussion within the IETF community and catalyze future work in this area. To this end, our strategy has been to a) develop the rationale for using SIP b) evaluate, at a high-level, a variety of SIP-based approaches, and c) identify, if appropriate, the future developments.

2. Terminology

See [[RFC3261](#)] and [[RFC2326](#)] for terminology.

In addition the following terms are defined:

Trick Play: Play, stop, pause, rewind and fast forward of streaming media.

<< others to be added when necessary >>

3. Controlled Streaming media applications and their characteristics

As IP-based broadband data services have continued to develop and expand, opportunities for streaming media applications have also proliferated and expanded beyond the traditional framework. This section describes several streaming media application use case scenarios. These scenarios illustrate the variety of conditions and environments in which streaming media applications need to operate.

3.1 Scope

The scope of applications for this document includes applications with the following characteristics: content-on-demand, streaming media, unicast-media streams, live or recorded content, ubiquitous access (any-device, any-access).

Explicit exclusions include non-streaming (e.g., downloaded media); though of interest this is out of scope for the present document.

3.1 Characteristics

For the purposes of this document, the term 'controlled streaming media application' represents the class of applications with the following characteristics:

- multiple servers that can be a source of content but showing up as a single stream at the client
- <<is this restrictive?>>
- one or more clients that receive the content

- the media stream(s) needs to be delivered isochronously, (most common case: the client intends to begin rendering the media before delivery is complete - less common but equally valid, the server does not have resources to buffer content until the client is ready to receive it, e.g., a live feed)
- a session exists between each client and server.
- the session is established, managed, and terminated through the use of a signaling protocol, in which control messages are exchanged (either directly or indirectly) between the client and the server. referred to as 'session signaling'
- the application supports media stream control. The client (or a proxy element acting on behalf of the client(s)) has the ability to manipulate the media stream (or other aspect of the application) via a signaling. This referred to as application-signaling (or media control signaling)

3.2 Examples of Controlled Streaming Media Applications

This section provides some examples of 'controlled streaming media applications'. The examples are by no means exhaustive, but are included here to illustrate the variety of applications and range of scenarios that fall within the scope of consideration.

- Content on Demand Entertainment Applications
include Video and Audio on demand
- Interactive Television Applications
- Remote Monitoring Applications
- Distance Learning Applications
- Personal Content Sharing Applications
- Applications involving Converged Services (video conference to the settop, web services to the settop etc.)

3.3 Aspects of Controlled Streaming Media Applications

Controlled streaming media applications are characterized by:

<<Each aspect needs a description>>

- pre-recorded versus live content
- use of network or personal digital video recorders (DVRs)
- types of media (audio/video) and encoding (and need for transcoding)
- location of clients & servers in the network
- behind NAT/FW
- transport operators/number of proxies traversed between clients and servers
- reachability of clients and Servers
- persistent reachability via host domain name, non-registered hostname/ non-persistent address

- locus of media control
- at client
- shared among clients (?)
- at centralized or network entity that serves one or more clients
- relationship between content provider and transport provider(s)
- best effort versus enhanced quality of service transport

3.4 Use Cases

Use cases are used with the purpose of clarifying the 'streaming media application' and to explore the space of applications. The objective is to

- clarify the frame / scope the discussion
- illustrate some of usage scenarios
- identify some of the key attributes that characterize these

Scenario 1 - Converged Services

A user is watching live TV. A VoIP call is incoming and the call information needs to be displayed on the TV. Without SIP the video stream and signaling stream need to be coordinated at the end point. With SIP the muxing of the two is greatly simplified. For example, based on user preference the network can issue a PAUSE of the stream and SEND the call information to the TV.

Scenario 3 - Remote video monitoring

If the user wants to watch a live content or request video on demand, the communication is generally established by RTSP, and the exchange of media is uni-directional. Instead of RTSP, SIP can also be used to setup the communication, which gives the advantage to be able to switch from a one-way monitoring communication to a bidirectional communication. Although SIP, HTTP or FTP can also be used instead of RTSP, RTSP has here the advantage to include trick play commands to remotely interact on the stream. This use-case clearly shows that making the choice to use SIP or RTSP leads to some advantages and disadvantages for each choice. Having a convergence between the 2 protocols would help to benefit from the advantages of the 2 protocols.

Scenario 5- Multicasting

While multicasting creates an interesting service offering for displaying content simultaneously on different end devices. Multicasting sessions can be described either by RTSP or SIP, but they do not allow for a full trick play solution. <<details to be provided >>

Scenario 5 - Video on Demand:

Consider a Video on Demand (stored video) service provided on a SIP based network. The VOD service is provided as a unicast session to an end user device from a server. The user contacts an application server (AS) to request a VOD service. The AS determines the video that user wants to watch, and then contacts the appropriate network element (NE) and requests it to reserve resources for the user and confirm back to the AS. The problem is here that until this point, the NE can not fully estimate and negotiate the media resources needed for the whole duration of the VOD. An example is in case of NAT/firewalls, when the NE can not know the IP-address/port on which the RTP stream should be streamed - the RTP IP-address/port will only be decided when the RTSP session kicks off. If the synchronization of SIP and RTSP is achieved, SIP can do all the session setup negotiations (replacing the RTSP SETUP method). RTSP will need to take on from the PLAY onwards. << As this is an "operator" requirement; may not be included in the future>>

<< More use cases are being developed and correspond to the requirements of the next section >>

4. Required capabilities

This section lists key requirements derived from the application use cases described in the previous section. The requirements are described in terms of the capabilities provided by a prospective solution.

4.1 Scalability

Any solution must be able to accommodate:

- Millions of clients and servers operating simultaneously
- An individual server may need to support thousands of simultaneous sessions.
- An individual client may need to support a handful of Simultaneous sessions.

4.2 Signaling Latency:

Because the use cases refer to live content the latency budget is important for the user experience. Hence the latency is divided in:

- Session Signaling Latency: this refers to the latency of the initial session setup as well as other session related events
- Application/media Control Latency: this refers to the end to end response for the media controls during the session

4.3 User identification, authentication and authorization

<<words to be added>>

4.4 Accounting, charging, and settlements

<<words to be added>>

[4.5](#) Server Location Discovery

<<words to be added>>

[4.6](#) NAT and Firewall Traversal

The solution must provide a way to traverse NATs and Firewalls. For example, when switching from a remote monitoring communication to a conversational communication, the NATs bindings should not be renegotiated.

[4.7](#) Session-based transport policy control

<<words to be added>>

[4.8](#) Extensible with respect to application control signaling

(Support many different application types)

<<words to be added>>

[4.9](#) Support media negotiation

The solution must provide a way to negotiate all media sessions (eg: conversational and streaming) as a whole, as described in the [RFC3264](#) with the offer/answer so that both parties can estimate the media resources involved in the session.

[4.10](#) Allow proxies:

[4.11](#) Support media negotiation:

Including per-flow/media QoS

<<words to be added>>

[4.12](#) Support Converged Services

<<words to be added>>

[4.13](#) Support lightweight clients

<<words to be added>>

[4.14](#) Support auto-configuration/installation

<<words to be added>>

[5.](#) Rationale for using SIP

As mentioned in the introduction, both SIP and RTSP are session setup and control protocols and they contain many overlapping capabilities besides just being similar in syntax and operation to HTTP/1.1, especially for session setup. While RTSP predates SIP and has been successfully deployed, in some networks like the Internet Multimedia Subsystem (IMS) it is now mandated to use SIP for session setup.

SIP is used for controlling conversational communications, but also for Presence and Instant-Messaging. SIP is defined in [[RFC3261](#)] and supports several extensions. SIP has the following properties:

- acts as a rendezvous protocol (with many capabilities)
- carries the session description protocol
- supports invitation to unicast or multicast sessions

Whitehead

Expires April 30, 2006

[Page 8]

RTSP is used for controlling streaming communications such as the delivery of webcam monitoring content, the delivery of Video On Demand, and, in some cases, the delivery of IPTV. RTSP is defined in [[RFC2623](#)] and has the following properties:

- acts as a lightweight rendezvous protocol
- supports trick plays and media control (pause/rewind/forward/...)
- carries the session description protocol
- supports invitation to unicast or multicast sessions

[5.1](#) Why SIP?

Amongst the advantages of using SIP is the large investment by both industry and operator that is reflected in the extensive deployment of SIP based services for multimedia especially for wireless services and Voice over IP (VoIP). Another advantage is the large number of standardization efforts surrounding it. This includes the IETF level (e.g.: preconditions, offer/answer ...), but also in other standardization bodies (e.g. 3GPP, TISPAN, ITU, PacketCable). Using SIP when it makes sense means that the work already done for SIP will not need to be repeated when new services are introduced.

For converged services that use both real time streaming and messaging capabilities, it would be desirable to eliminate the redundancies presented by the two protocols especially to overcome the delays and added complexities of satisfying network requirements when both are being used for session setup.

End device aspects also need to be considered. It has been common in IPTV to push the intelligence to the servers and clients. It is the role of the end device to blend the services when required. While this approach is sensible devices with large memory, CPU and power, it may not be true for the range of devices currently being deployed for future services or for some legacy devices that include video settop boxes. Using a SIP-based solution may allow to push some of the intelligence back in the network and allow the support of thin clients.

In addition, SIP:

- Enables centralized AAA for registration and connection establishment
- Supports concept of location service
- Supports peer to peer signaling

5.2 Why not SIP?

The disadvantages of SIP include:

- the large body of video applications using RTSP, and thus the development needs in these video servers in case of adding the SIP protocol
- the latency issues that reflect the number of proxies on the signaling path: SIP isn't appropriate for "live" real-time controls unless the network allows giving priority to SIP controls
- the high degree of flexibility in SIP introduces significant complexities not needed if given objectives are just video transmission
- no reliance on network entities for user applications

6. Solution Options

<< This section contains all the solutions that were submitted by the contributors; in a future version they will be filtered against the requirements>>

Figure 1 presents the solution space. Block A shows current SIP based solutions and block B current dual stack solutions. Figure 1 is structured as to show the level of changes to current approaches with top solutions proposing little or no changes going down to more involved proposals. Any proposed solution must allow synchronizing SIP and RTSP sessions and providing the ability to "setup" the session using SIP and either use SIP or RTSP for session control. Examples include:

- Block A:
 - SIP only
- Block B:
 - SIP with RTSP/RTSPv2
 - SIP with MRCP
 - SIP with MSRP
 - SIP with BFCP

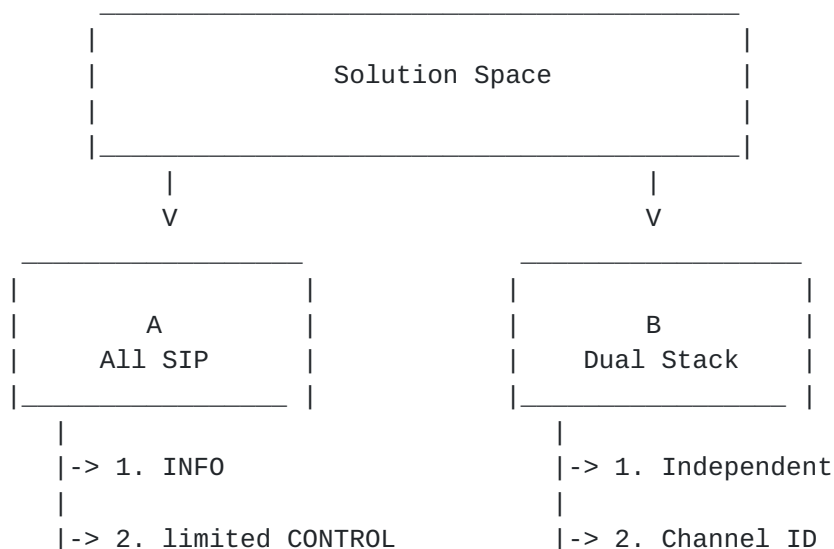


Figure 1. Solution Space
Expires April 30, 2006

In this section the generic approaches are further detailed.

6.1 SIP-only Options

The first approach would be to completely eliminate the need for RTSP and use SIP for all the setup and control functionalities. In this approach both the session and media control is performed via SIP. There - can be multiple ways of achieving this as can be seen below. These include:

- new SIP headers (eg SIP-MEX with new SIP headers in INVITE/UPDATE)
- new SIP bodies (new XML body in SIP INFO message)
- new SIP methods
- additional extensions in existing SIP bodies (eg: SDP attribute)

<<to be added>>

Figure 2: SIP-only only options

6.1.1 Use of SIP INFO

The session setup via the INVITE establishes a context for the control messages a session identified by the call-id. The INVITE includes a REQUIRED header that names the "trick-plays" extension package that indicates that trick-play controls would be delivered from the client to the sip-server via SIP-INFO.

Most of the SIP devices run over the UDP for call establishment. The xml-types streaming/media controls over the SIP using INFO approaches would be required for a full media control solution. While SIP/Info streaming control for announcement and recording is appropriate, but for real time stream control (and trick plays) it may not be enough, because of the big delays when, for example an Application Server is involved in the DTMF detection and creation of the appropriate control to the media server.

As long as video servers do not support SIP, at the SIP-server this means a back-to-back connection: one side is the SIP session to the client, the other side is a RTSP session to the video server for example. Then any INFO message sent from the client to the sip-server would be received, translated into its RTSP equivalent and sent to the video server.

6.1.2 Media controls in SIP Control extensions

This solution consists in defining new SIP extensions to perform the media controls (pause, rewind, and forward). For example, a new SIP REWIND command with some new headers for some options (e.g. speed).

The main drawback is a potential problem of delays: in case there are multiple proxies in the route of the SIP messages there may be a delay,

for example between the time when the user performs a pause command and the time when the pause command is really performed at the media level.

Whitehead

Expires April 30, 2006

[Page 11]

6.1.3 New SIP headers extensions

This approach defines extensions in the SIP protocol, especially in the headers, in order to provide SIP with the necessary RTSP functionality.

The key features of the SIP-MEX are:

- Re-INVITE or UPDATE is used as PAUSE, RECORD and PLAY of RTSP
- SIP headers are extended with RTSP media control headers (Scale, Speed, Range)

One known implementation example is "SIP with Media Streaming extensions", which in addition also has additional enhancements such as SIP subscription/notifications to receive session modifications, and also additional URI parameters for addressing streaming resources.
<<more to be added>>

6.1.3.1 Advantages

The generic advantages of the SIP only solution include:

- Single protocol stack on the client
- Unified signaling (converged networks)
- Flexibility
- Integration with non video services

6.1.6 Disadvantages

The generic advantages of the SIP only solution include:

- SIP delay (propagation via the network)
- May need RTSP on the server side
- Video/media server developments
- New Supported and Require header are needed

6.2 Dual protocol Options

The second approach would be to use SIP for initial session setup, and use another protocol for various control functionalities (play/pause/rewind/forward). In this approach separate stacks are kept, one SIP stack and one RTSP (or other) stack, with the switching between the RTSP and the SIP is left to application logic. This solution continues to use the RTSP (or other) controls to reduce the latency of trick plays and channel controls due to SIP propagation. RTSP2 (the last draft [[IDRTSP](#)]) defines RTSP to use TCP as transport protocol.

The client application includes a SIP stack and a 'lightweight' media control stack. The lightweight stack corresponds to software that supports media control operations and a simplified 'setup' mechanism. The media server supports both SIP and RTSP. The SIP component is used to negotiate media sessions using SIP for session management. The RTSP component is used for handling media control messages in RTSP and can also be used to manage sessions established using classical RTSP

DESCRIBE/SETUP procedures. While the media server could be split in a

Whitehead

Expires April 30, 2006

[Page 12]

SIP and a RTSP component the integrated approach presents numerous advantages in terms of coupling and reduced complexity as well as availability.

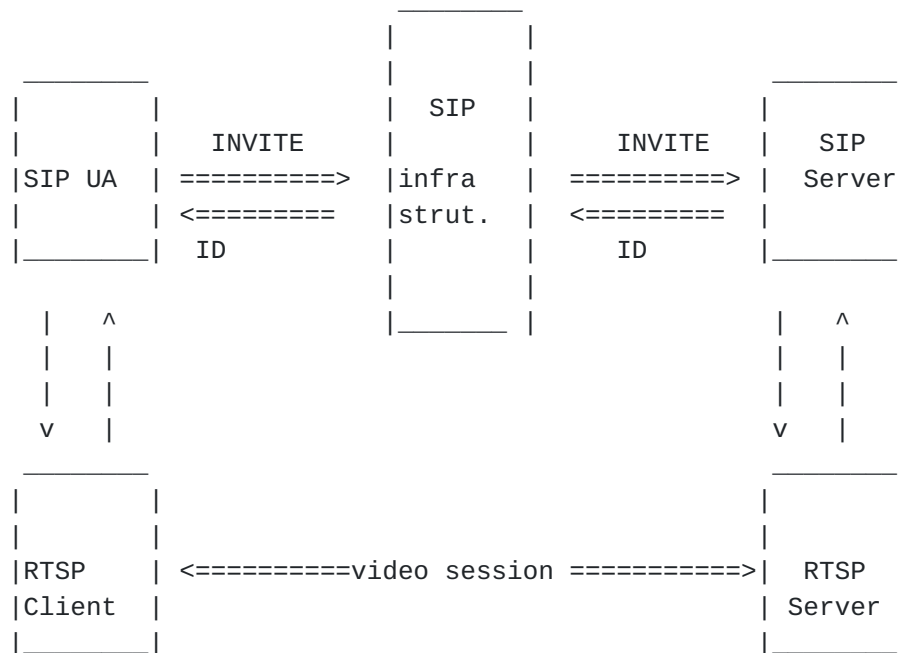


Figure 2: Channel Identifier Solution

The combination of SIP/MRCP2&RTSP provides a good approach for media server control. SIP for session establishment and MRCP or RTSP for the streaming control and media services. There could one common technique for synchronization between SIP and the other protocol. It may be done using the SDP (for example SIP/MRCP2) and some form of ID.

6.2.1 SIP/RTSP

In this solution the channel identifier is passed from the SIP session to the RTSP session (in a push or pull manner. This is an opaque identifier. Thus the 2 protocols could stay as independent as possible, and the modifications would be limited to the minimum, which makes sense to limit the needed evolutions in legacy systems.

<<needs more details>>

6.2.2 SIP/MRCP2

Another proposed solution that isolates the session initiation from the media stream control is based on the SIP/MRCP2 draft [[IDMRCP](#)]. Note that MSRP and BFCP are alternative candidates to MRCP2. It may be done using the SDP (for example SIP/MRCP2 - see the Appendix).

6.3.3 Common elements

Both SIP/RTSP and SIP/MRCP2 need minor changes to the RTSP2 to support Channel-Identifier.

The MRCPv2 approach can be used with:

a=channel

for channel synchronization and negotiation.

m = application 32416 TCP/RTSP

should be used for the RTSP support.

Regarding the RTSP services:

a=resource:xxx

may be defined. There is no need for a DESCRIBE message to be sent, because all the negotiation is done via SIP/SDP.

The RTSP:SETUP message, in some cases it may be not needed. The PLAY command may be sent directly with Channel-Identifier.

6.2.5 Advantages

The major advantage of this approach is to provide one common interface to the Media Server. The SDP may include all needed information about the interfaces that may be required and supported for a specific call (One SDP with all needed information). All services may be negotiated at the INVITE stage. Another advantage is one resource allocated at the Media server (Implementation dependent) to support various services. In addition, a distributed architecture is supported as multiple media servers may participate in one session according to requested service.

6.2.6 Disadvantages

These solutions need changes to RTSP.

7. Analysis

<<This section will be included in future versions; it will evolve based On feedback from v0; depending on how much analysis is in [section 6](#) we may want to merge this section into a final analysis summary and recommendations section>>

8. Recommendations

<< This section will be included in future versions; it will depend on feedback from v0>>

9. IANA Considerations

The RTSP 'encoding format' and the new media attributes may need to be registered.

10. Security Considerations

No rogue 3rd party should be allowed to get the token and use it to set up an un-authorized RTSP session. One solution is to associate the token to the known RTSP session endpoints (negotiated in SIP) so the server could reject any token that came from an endpoint that didn't match what it expected. <<can that be spoofed?>>

11. Acknowledgements

Many thanks to those who provided valuable inputs for this document namely Darren Loher, C. Steck, Osher Hmelniczky, Jonathan Rosenberg, David Ress, Ravishankar Shiroor, Martti Mela, Xupei Li and the members of the sip-rtsp mailing list.

12. Change History

None for now.

13. References

13.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [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.
- [RFC2326] Schulzrinne H., Rao, A., Lanphier R., "Real Time Streaming Protocol (RTSP)", [RFC 3261](#), April 1998.
- [IDRTSP] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., Narasimhan A., "Real Time Streaming Protocol 2.0 (RTSP)", October 2005.
- [IDSDP] Hautakorpi, J. and G. Camarillo, "The SDP (Session Description Protocol) Content Attribute", July 2005.
- [IDBFCP] Camarillo, G., Ott, J., and K. Drage, "The Binary Floor Control Protocol (BFCP)", November 2005.
- [IDSDPBFD] Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", November 2005.
- [RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [IDMRCP] Shanmugham, S. "Media Resource Control Protocol Version 2 (MRCPv2)", [draft-ietf-speechsc-mrcpv2-05](#), October 2004.

13.2 Informative References

[RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", [RFC 2396](#), August 1998.

14. Author's Address

Steven Whitehead
Verizon Laboratories Inc.,
40 Sylvan Road
Waltham MA 02451
781-466-4072
steven.d.whitehead@verizon.com

Marie-Jose Montpetit
Motorola Connected Home Solutions
55 Hayden Avenue 1st floor
Lexington MA 02421
781-372-4251
mmontpetit@motorola.com

Xavier Marjou
France Telecom
Rue Pierre Marzin
Lannion 22300
France
xavier.marjou@francetelecom.com

15. 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.

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

16. 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 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.

17. Copyright Statement

Copyright (C) The Internet Society (2005). 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.

18. Acknowledgment

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

19. Appendix

MRCP based Solution

MRCPv2 defines SDP changes at the invite message:

C->S: INVITE sip:mresources@server.example.com SIP/2.0

```
m=application 9 TCP/MRCPv2
a=setup:active
a=connection:new
a=resource:speechsynth
a=cmid:1
m=audio 49170 RTP/AVP 0 96
a=rtpmap:0 pcmu/8000
a=recvonly
a=mid:1
```


S->C: SIP/2.0 200 OK

m=application 32416 TCP/MRCPv2
a=setup:passive
a=connection:new
a=channel:32AECB234338@speechsynth
a=cmid:1
m=audio 48260 RTP/AVP 00 96
a=rtpmap:0 pcmu/8000
a=sendonly
a=mid:1

And each MRCP request sends

C->S: MRCP/2.0 44 XXX
Channel-Identifier:32AECB23433802@speechsynth

