Network Working Group	S. Whitehead	
Internet-Draft	Verizon Laboratories Inc.	
Intended status: Informational	M. Montpetit	
Expires: November 13, 2008	Motorola	
	X. Marjou	
	France Telecom	
	S. Ganesan	
	Motorola	
	J. Lindquist	
	Ericsson	
	May 12, 2008	

Media Playback Control Protocol Requirements
draft-whitehead-mmusic-sip-for-streaming-media-05

Status of this Memo

Abstract

The media playback control functionality controls the delivery of streaming media by the means of commands like pause, fast forward, fast rewind, slow forward, slow rewind. This document presents some of the requirements for a media control protocol that does not contain any session setup semantics in it.

Table of Contents

---

1. Introduction [TOC](#)

This document defines the requirements for a media control protocol distinct from the session control protocol for Content on demand media streams.
Historically media stream delivery has been controlled by RTSP as both the session set up protocol as well as the media control protocol. RTSP [RFC2326] (Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)," April 1998.) and its successor [RFC2326bis] (Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)," November 2007.) define semantics for both session setup as well as media control, with commands like pause, Rewind etc.
Similarly SIP has been the protocol of choice for end to end session establishment and rendezvous [RFC3261] (Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol," June 2002.). These

functionalities in the context of conversational services has been the main raison d'etre and forte of SIP.

There exist environments, circumstances and use cases where it is desirable to use SIP for session establishment and rendezvous, while retaining RTSP's capabilities for media and play back control. Such circumstances are increasing in number given that user agents with wide ranging capabilities are become much more common in deployments. Good Integration with existing RTSP deployments is also desirable under these conditions.

Section 3 (Use Case Scenarios) describes some use cases that motivate this work.

Section 4 (Requirements) lays out the requirements for a media control protocol, ideally some form of lightweight RTSP without its session establishment semantics, that can be used with a session establishment and rendezvous protocol.

---

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 [RFC2119] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

---

3. Use Case Scenarios

The scope of scenarios 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).

While of interest, non-streaming media applications, such as downloaded media services, are outside the scope of this document.

3.1. Characteristics

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

*Multiple servers that can be a source of content but showing up as a single muxed stream at the client.

*One or more clients can receive the content.

*The media stream(s) needs to be delivered isochronously, in the
 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 source (e.g. server or peer) and
 destination (e.g. client or peer).

*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 source and
 the destination referred to as 'session signaling'.

*The application supports media stream control. The client(s), 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 signaling. This is referred to as media control
 signaling (or application signaling).

3.2. Use Case Descriptions

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.
Use cases are used with the purpose of clarifying the 'streaming media
application' and to explore the application space. The objectives are
to:

*Clarify the frame / scope the discussion.

*Illustrate some of usage scenarios.

*Identify some of the key attributes that characterize these use
 cases.

3.3. Server Control of Streaming Session

During a streaming session, the operator may want to redirect or move pending sessions in order to for e.g., upgrade the server. Therefore, the server will initiate a session redirect. Another use case of Server control operation is when the server decides to initiate a session to the user, based on user preconfigured-settings (e.g. reminders). Further, sessions that are indefinitely paused by the client need to be terminated and server resources reclaimed at the operators discretion. This would also be true for sessions where the client may have become unresponsive.

3.4. Remote Access to Private/Firewalled Video Content

In this use case, the user, while not on his home network, wants to access content that is stored on his personal or home network, e.g., a pre-recorded show on a PVR device, or a monitoring camera at his or her home location that is capable of providing a live feed as well as record it locally as a PVR asset.
In the case of a live monitoring camera in a home network, the user wishes to transition from watching a live stream of the feed to being able to move backwards and forwards using media control commands on the stored content of the monitoring device. This translates logically to transitioning from watching live TV programming to Time shifted TV or PVR type of viewing. Being able to do it in the context of the same session is desirable.
In the case of a home network based PVR, it is more than likely that the home gateway is not set up to be an inbound device for various session setup requests to enable the external client to traverse the protocol unaware IP NAT device commonly found at the edge of home networks.
In both these cases, the video server is behind a firewall. In the first case, the transition from one mode (live feed) to another (COD), would entail multiple messages for staying within a session. Also, the client being exterior to the firewall, needs to establish a TCP connection from "out" to "in". RTSP as in stands currently does not deal with these adequately. A rendezvous capable protocol like SIP could provide this in addition to client identity and location.

3.5. VOD services that requires resource or QOS-guarantees

Consider a Video on Demand (stored video) service provided as a unicast session to an end user device from a server. The user requests a VOD movie. If the operator uses a network proxy to request and guarentee QoS for the delivery of the movie from the server to the client, currently RTSP does not provide the means of guarenteeing that all subsequent messages that form part of the RTSP session will go through the network

proxy that manages the QoS. In this particular use case, an end to end session setup and management protocol would be helpful.

3.6. Intelligent selection of media encoding

A user orders content to be delivered to its current device. The content could exist in different format (e.g. standard definition or high definition) or encoding (e.g. MPEG2, MPEG4, ...). In addition the device can be located behing different types of access networks, which implies bandwidth constraints. The selection of media encoding can be adjusted to accomodate these multiple characteristics.

3.7. Voice/video mailbox

The control of the server may be extended to a voice/video mailbox system. In addition to controlling the playing of the messages and fast forward or rewind similar to content on demand with a couple of additional commands to delete or save messages it is possible o have access to a mailbox system. THe mailbox may be located in the home or in the network accessed from the home. If the mailbox is at home then it is possible to remotely access to the messages.

3.8. Motion Detection

Motion-detecting or pattern-matching cameras may need to call a human when motion/pattern is detected and may also have a buffer, which allow the human to place pause/rewind commands.

3.9. Video Subscriptions

A user subscribes on a webpage to get all new local high-school football games, or Voipsa blue-box podcasts. He wants his DVR/TV to pop up with the option to play them immediately, or record them to DVR, or neither. The high-school football game is not available from cable TV, so the DVR doesn't have a schedule to know when to get it by itself. Of course this could be done with an off-line indication, such as email, but it would be nice if his DVR didn't need to support email.

---

4. Requirements

This section outlines the key requirements that need to be satisfied in
order to have a media control protocol acting as a control stream within
a multimedia session.

REQ-1 The media control protocol must support commands such as
play, pause, rewind, forward, fast rewind, fast forward, slow
rewind, and slow forward.

REQ-2 It must be possible to negotiate the media control protocol
of a media stream.

REQ-3 If the media control protocol does not apply to all media
streams of a given session, it must be possible to indicate the
specific media streams that are under the scope of the trick-play
control protocol.

REQ-4 The media control protocol must allow for asynchronous media
event notifications (e.g.: end-of-stream)".

REQ-5 The protocol SHOULD work over TCP.

REQ-6 The media stream, or media control server, to be controlled
by the client may be located in the network or in the home
network.

REQ-7 The media control protocol shall consider additional commands
not available in RTSP to control the media in the server.
Examples of such commands are deletion or saving of voice
messages.

---

5. Considerations for session control protocol

This section raises a number of areas that need consideration while
developing new standards for the use cases:
1. RTSP protocol assumes a media server is located in the network and
not in the home. The session protocol shall be possible to establish a
relationship that allows for control of media resoruces in both the home
and network.
2. The session protocol shall be able to handle NAT and not affect the
call flows being defined.
3. If assuming SIP for session protocol there is a well accepted
architecture defined called IMS, IP Multimedia Solution, which is
accepted by a number of standard organizations like 3GPP, ETSI TISPAN,
ATIS, etc. There is a number of services have been defined using the
architecture like telephony, push to talk, presence, messaging, chat and
IPTV.

4. In conjunction with IMS there is a resource and admition control architecture called RACS defined in ETSI TISPAN which helps ensure QoS for services defined over IMS and addresses reuse of network resources. What is of special interest is bandwidth reservation is addressed for unicast and multicast media streams as well as handling of multicast addresses used for IPTV.
5. IMS also provides additional authentication mechanisms which allow alternatives for HTTP Digest like the Authentication and Key Agreement mechanism (AKA).
6. The session protocol shall allow server initiated control of streaming sessions such as server-initiated session terminations. RTSP TEARDOWNs are from client to server only.
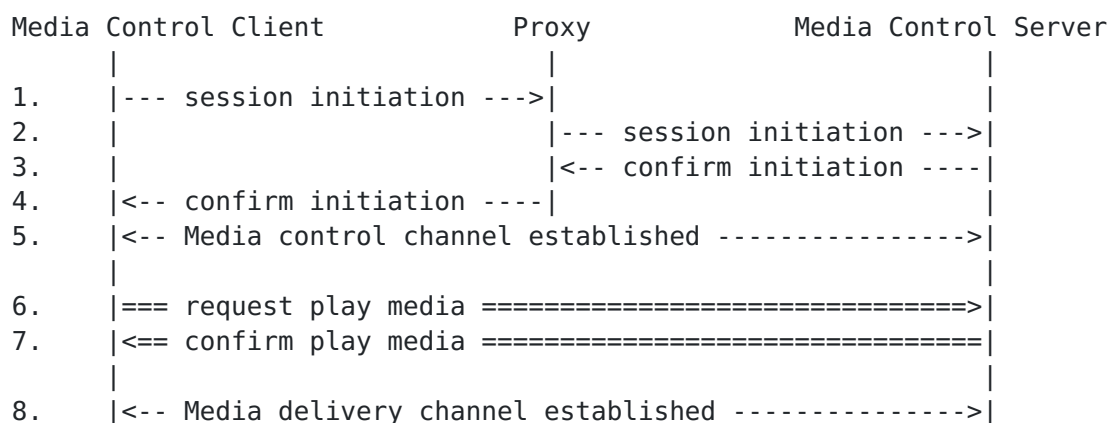
---

6. Call Flows of Use Cases

6.1. Content on Demand Session Initiation

For content on demand the session initiation established two media streams, one for media control channel and one for media delivery channel between the entities Media Control Client and Server. All required information for establishing the two channels are conveyed in the session initiation.
The proxy acts as back-to-back user agent for the session control. The proxy may open pin-holes for the media control channel and protocol. The proxy is used to protect the core network which the Media Control Server is located.

```
    Media Control Client              Proxy            Media Control Server
         |                              |                         |
    1.   |--- session initiation --->|                           |
    2.   |                              |--- session initiation --->|
    3.   |                              |<-- confirm initiation ----|
    4.   |<-- confirm initiation ----|                             |
    5.   |<-- Media control channel established --------------->|
         |                              |                         |
    6.   |=== request play media ==============================>|
    7.   |<== confirm play media ===============================|
         |                              |                         |
    8.   |<-- Media delivery channel established -------------->|
```

The Media Control Client or Server do not need to be strickly located in a home for the Client and the network for the Server. The roles may reversed. The Client may be located in the network as a remote user attempting to access the content in the home; the Server is then located in the home.

Other use cases compared to content on demand may be supported that
follow theses sequences. For example voice/video mailbox may be
supported.

6.2. Content on Demand Session Termination

When content on demand is terminated either the media control client or
server may initiate a session termination. Pin-holes that may previously
have been established are closed.

```
Media Control Client               Proxy             Media Control Server
          |                          |                          |
1.        |--- session termination -->|                          |
2.        |                          |--- session termination -->|
3.        |                          |<-- confirm termination ---|
4.        |<-- confirm termination ---|                          |
```

Alternatively the server may terminate the session.

```
Media Control Client               Proxy             Media Control Server
          |                          |                          |
1.        |                          |<--- session termination --|
2.        |<--- session termination --|                          |
3.        |--- confirm termination--->|                          |
4.        |                          |--- confirm termination -->|
```

6.3. Activation of trick play of linear TV Session Modification

The first 5 steps setup linear TV and a multicast media delivery
channel. If proxy is aware of the bandwidth limitations for the client
it may reserve the required bandwidth for the session. Note that
parallel sessions may be established for the same client to convey
multiple linear TV channels at the same time.
When the user pauses live TV the same session is used to modify the
media requirements from a multicast media channel to a media control and
unicast media channels. If session modification is successful then the
live TV is paused. If unsuccessful then the linear TV is maintained
without affecting the use viewing experience. By performing the action
within the same session network resources are not released when
transitioning between multicast and unicast otherwise there is a risk
that resources are not anymore available if trying to reestablish the
previous session.

```
    Media Control Client              Proxy              Media Control Server
          |                             |                             |
1.    |--- session initiation --->|                             |
2.    |                             |--- session initiation --->|
3.    |                             |<-- confirm initiation ----|
4.    |<-- confirm initiation ----|                             |
5.    |<-- Multicast media delivery channel established ----->|
          |                             |                             |
6.   User pauses live TV
7.    |--- session modification ->|                             |
8.    |                             |--- session modification ->|
9.    |                             |<-- confirm modification --|
10.   |<-- confirm modification --|                             |
11.   |<-- Media control channel established --------------->|
12.   |=== request pause media ============================>|
13.   |<== confirm pause media =============================|
14.   |<-- Unicast media delivery channel established ------>|
```

6.4. Deactivation of trick play of linear TV Session Modification

When the user switches TV channels or catches up with live TV then the
session modification is performed and media requirements are changed
from a media control and unicast media channels to a multicat media
channel. Note that if user chooses to end viewing session termination is
performed directly with modification.

```
    Media Control Client              Proxy              Media Control Server
          |                             |                             |
1.    |<-- Unicast media delivery channel ongoing ----------->|
          |                             |                             |
2.    |--- session modification ->|                             |
3.    |                             |--- session modification -->|
4.    |                             |<-- confirm modification ---|
5.    |<-- confirm modification --|                             |
          |                             |                             |
6.    |<-- Multicast media delivery channel established ------>|
```

7. Security Considerations

T.B.D.

8. IANA Considerations   [TOC]

This document has no actions for IANA.

9. Acknowledgements   [TOC]

Thanks to Christer Holmberg from Ericsson, Priya Rajagopal from Motorola
, Mikhael Said from France Telecom, Dan Wing from Cisco, and Hadriel
Kaplan from Acme Packet.

10. References   [TOC]

10.1. Normative references
                                                               [TOC]

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

10.2. Informative references
                                                               [TOC]

[RFC2326]	Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, April 1998 (TXT).
[RFC2326bis]	Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)," draft-ietf-mmusic-rfc2326bis-18 (work in progress), November 2007.
[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).

Authors' Addresses
                                                               [TOC]

	Steven Whitehead
	Verizon Laboratories Inc.
	40 Sylvan Road

	Waltham, MA 02451	
	USA	
Email:	steven.d.whitehead@verizon.com	
	Marie-Jose Montpetit	
	Motorola	
	900 Chelmsford Street	
	Lowell, MA 01851	
	USA	
Email:	mmontpetit@motorola.com	
	Xavier Marjou	
	France Telecom	
	Rue Pierre Marzin	
	Lannion, Brittany 22307	
	France	
Email:	xavier.marjou@orange-ftgroup.com	
	Sam Ganesan	
	Motorola	
	80 Central Street	
	Boxborough, MA 01719	
	US	
Email:	sam.ganesan@motorola.com	
	Jan Lindquist	
	Ericsson	
	Tellusborgsvaegen 83-87	
	Hagersten, Hagersten 12637	
	Sweden	
Email:	jan.lindquist@ericsson.com	

Full Copyright Statement

Intellectual Property